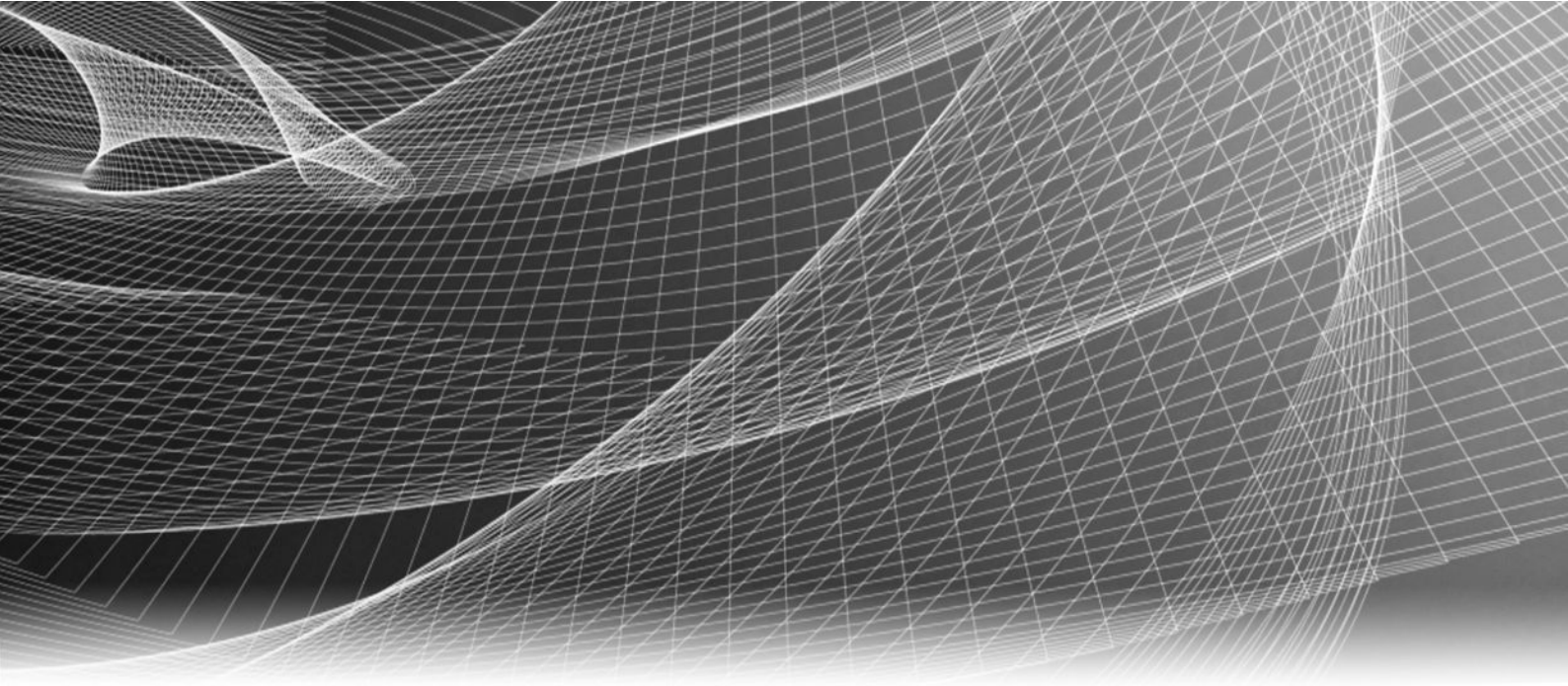


EMC Solutions Enabler Array Management

Version 8.1.0

CLI User Guide

REVISION 01

EMC²

Copyright © 2015 EMC Corporation. All rights reserved. Published in USA.

Published September, 2015

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 EMC Online Support (https://support.emc.com).

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

CONTENTS

Figures 19

Tables 21

PREFACE 23

Part 1 Setting up and Configuring Storage Arrays 27

Chapter 1 Introduction 29

Solutions Enabler overview 30
SYMCLI help options 30
Environment variables 31
Command input and output tips 33
Version compatibility information and restrictions 33

Chapter 2 Discovery 37

Discovery overview 38
Discovery syntax options 38
Verify host configuration database is synchronized 38
Scan for new devices 39
Connectivity authorization 39
Configuring virtual disk mapping 40
Display virtual disks as array devices. 41
Host configuration database file 41
Database file location 41
Database file lock 42
Changing the current database file name 42
Database location in client/server mode 42
Database access modes 42
Command modes: Online and offline 43
Database configuration information 43
Configuration data synchronization 43
Update configuration information (sync operation) 44
Inhibit Database synchronization 44

Chapter 3 User and Group Authorization 47

User and group authorization overview 48
User roles 48
Assign multiple user roles 49
User authorization rules 49
User ID formats 50
Authorization settings management 51
Current username identification 51
User authorization 52
User authorization enforcement 53

Modify user authorization role mappings 54
User authorization command file usage 54
Preview and commit user authorization actions 55
Backup the authorization database 55
Restore the authorization database 56
Monitor authorization 56
List user roles, names, and components 56

Chapter 4 Host-based Access Control 59

Host-based access control overview 60
Access Control database 60
Securely identifying hosts with access IDs 61
Create or modify access control data 61
Applying access control operations 62
Minimum access control configuration 62
Initial ACL setup 63
Create and manage access groups 64
Verify administrative authority 64
Create an access group 64
Add host access IDs to a group 64
User access IDs (PINs) for AdminGrp 65
Edit and manage an access group 66
Create and manage limited access to access pools 67
Verify administrative authority 68
Create an access pool 68
Add devices to access pool 68
Edit and manage access pools 69
Create and manage access control entries 69
Verify administrative authority 70
Grant permissions to access group 70
Available access control permissions 70
Grant BASE permissions to a group 73
Grant SRDF permissions to a group 74
Grant BCV and SDR permissions to a group 74
Grant BASE permissions to a non-registered host 74
Remove permissions from access group 75
Obtain access control information 75
List access control information 75
Show access control information 75
Obtain host access ID 76
Verify a locked access control session 76
Release locked access control sessions 76
Access control strategies 76
Default configuration: all permissions to users and hosts 77
Manage default access 77
Setting up access control for TimeFinder devices (5977 and higher) 79
Setup access control for TimeFinder/Snap devices (Enginuity 5876 or lower) 80
Setting access control for SRDF devices 80
Setup access control for RecoverPoint Splitter (Enginuity 5876 or lower) 80
Set access control for FAST/Optimizer (Enginuity 5876 or lower) 81
Symmetrix Management Console/Unisphere setup strategy 81
Setup access control for Symaccess 81

Back up and restore the access control database 81
Create an access control database backup file 81
Restore access control database with backup file 82

Chapter 5 Grouping Devices 83

Device group overview 84
Device groups 84
Group name services 84
Names of device groups and devices 84
Device group types 85
Device lists 85
Device restrictions 86
Create a device group 87
Add devices to a device group 88
Set controls on Celerra devices 91
List devices in a device group 92
List all device groups 92
Show device group details 92
Export and import device lists 93
Rename device groups 95
Rename logical device names 95
Move a device between device groups 95
Move all devices between device groups 96
Copy devices between device groups 97
Copy devices from device group to storage group 97
Remove a device from a device group 98
Remove all devices from a device group 98
Delete a device group 99
Perform control operations on device group devices 99
Storage groups 100
Create storage groups 101
Create devices and add to storage groups 103
Add existing devices to storage groups 104
Add child storage groups to existing storage groups 105
Modify storage group properties 105
Standalone storage groups and cascaded groups 109
Remove devices from a storage group 112
Remove child storage groups 113
Host I/O Limits for storage groups (5876 or 5977) 114
Copy devices from a storage group to a device group 117
List storage groups 118
Show storage group details 121
Show cascaded storage group details (Enginuity 5876 or HYPERMAX OS 5977) 121
Export and import device lists 123
Rename a storage group 126
Move devices between storage groups 126
Copy a device between storage groups 128
Copy all devices between storage groups 129
Copy devices from a storage group to a device group 130
Delete a storage group 131
Perform control operations on storage group devices 131
Composite groups 132
Composite group device members 133
Logical device name support 133

Device group members within composite groups 133
Logical names of devices within a composite group 134
Composite group types 134
SRDF consistency groups 135
Create a composite group 135
Export a composite group to file 137
Delete a composite group 138
Import a composite group 138
Add standard devices to a composite group 138
Move and copy devices in composite groups 140
Remove a standard device from a composite group 141
Rename a composite group 143
Set SRDF group attributes 143
Composite group creation and output 143
Display composite group information 144
Show composite group details 144
Show composite groups associated with a device group 146
Perform control operations on composite group devices 146
GNS repository 148
Automatic upgrade 149
Shared group definitions with GNS 149
GNS updates to group definitions 150
GNS setup 152
GNS in a multihost environment 153
GNS user authentication 153
Start the GNS daemon 154
Restart the GNS daemon 154
Manage the GNS daemon 155
Access groups in offline mode 155
View and release GNS daemon external locks 156
GNS daemon log file 156
GNS daemon options file 157
Remotely mirror device group definitions 159
Identify GNS groups 161
GNS backup and restore mechanism 162
Manual backup to the GNS database 162
Manual restore from the GNS database 163
Troubleshoot GNS 163
List GNS data about groups 164
Show GNS data about groups 164

Chapter 6 Fully Automated Storage Tiering 167

Fully Automated Storage Tiering (HYPERMAX OS 5977) 168
Storage Resource Pool management 169
FAST information reporting 171
Fully Automated Storage Tiering (Enginuity 5874 through 5876) 180
FAST and FAST VP comparison 181
FAST configuration 181
Manage FAST tiers 182
FAST policy management 191
FAST-related storage groups 196
FAST controller management 202
FAST information display 214

Chapter 7 Manage Configuration Changes 223

Configuration change management overview 224
Configuration change session rules 224
Symconfigure command execution formats and options 224
Verify and check sessions 226
Abort configuration session 227
Configuration change guidelines 227
Verify viable array configuration 228
Check for free physical disk space 228
Stop I/O activity on affected devices 228
Mixed array environments 229
Blocked operations during active change session (Enginuity 5876 or lower) 229
Update host with device mapping information 229
Supported configuration operations 230
Set array attributes 234
View the array metrics 235
Set SLO name (HYPERMAX OS 5977 or higher) 235
Rules and restrictions for set/reset SLO name 236
External disk group management 236
Create devices (HYPERMAX OS 5977 or higher) 236
Restrictions when setting device names (HYPERMAX OS 5977) 238
Restrictions when creating devices (HYPERMAX OS 5977) 239
Modify device capacity (HYPERMAX OS 5977 or higher) 240
Create devices (Enginuity 5773 through 5876) 240
Multi-session device creation (Enginuity 5773 through 5876) 240
Create one or more devices (Enginuity 5773 - 5876) 242
Restrictions when setting device names (Enginuity 5773 - 5876) ... 247
Restrictions when creating devices (Enginuity 5773 - 5876) 247
Copy devices 249
Copy a similar device 249
Metadevice management (Enginuity 5773 - 5876) 251
Metadevice size requirements (Enginuity 5773 - 5876) 252
Meta performance (Enginuity 5773 - 5876) 252
Metadevice data addressing (Enginuity 5773 - 5876) 252
Metadevice creation methods (Enginuity 5773 - 5876) 253
Form metadevices (Enginuity 5773 - 5876) 253
Form a metadevice (Enginuity 5773 - 5876) 254
Create RDF metadevices (Enginuity 5773 - 5876) 254
Create AS/400 metadevices (Enginuity 5773 - 5876) 255
Meta configuration restrictions (Enginuity 5773 - 5876) 255
Add members to a concatenated metadevice (Enginuity 5773 - 5876) 256
Add members to a striped metadevice (Enginuity 5773 - 5876) 257
Remove metamembers (Enginuity 5773 - 5876) 257
Dissolve a metadevice (Enginuity 5773 - 5876) 258
Metadevice data preservation (Enginuity 5773 - 5876) 258
Configure thin metadevices (Enginuity 5773 - 5876) 259
Thin BCV metadevice guidelines (Enginuity 5875 - 5876) 259
Bind a thin metadevice to a pool (Enginuity 5773 - 5876) 260
Convert devices 260
Device conversion restrictions 262
Valid thin device conversions 262
Convert a thin device 263
Thin device recommendations to maximize I/O activity 263

Device group invalid warning (Enginuity 5773 - 5876) 263
Convert a metadevice (Enginuity 5773 - 5876) 264
Supported thin metadevice conversions (Enginuity 5773 - 5876) .. 264
Set device attributes 265
Device attribute values 265
Set device geometry on Solaris hosts (Enginuity 5773 - 5876) 267
Set device geometry (Enginuity 5773 - 5876) 268
Show device geometry 269
Set device identifiers 270
Identifier exclusions 271
View device identifiers 271
Delete devices (HYPERMAX OS 5977)) 271
Delete devices (Enginuity 5773 - 5876) 272
Device reservation management 273
Reserve devices 274
Commit changes on reserved devices 275
View reserved devices on an array 275
View details on reservation ID 275
Release reserved devices 276
Device pool management (HYPERMAX OS 5977) 276
Device pool management (Enginuity 5773 - 5876) 276
Device pool attributes and rules (Enginuity 5773 - 5876) 277
Create a device pool (Enginuity 5773 - 5876) 277
Delete a device pool (Enginuity 5773 - 5876) 278
Add devices to a pool (Enginuity 5773 - 5876) 278
Remove devices from a pool (Enginuity 5773 - 5876) 278
Enable or disable devices in a pool (Enginuity 5773 - 5876) 279
Rename a device pool (Enginuity 5773 - 5876) 279
Monitor a device pool (Enginuity 5773 - 5876) 280
View a device pool (Enginuity 5773 - 5876) 281
Map devices 281
Obtain list of addresses 283
Map CKD devices (Enginuity 5773 - 5876) 283
Map a range of devices (Enginuity 5773 - 5876) 283
Copy the mapping of a range of devices (Enginuity 5773 - 5876) ... 284
Assign PAV alias addresses to mapped devices (Enginuity 5773 - 5876) 284
Remove PAV alias addresses from mapped devices (Enginuity 5773 - 5876) 285
Remove PAV alias addresses from CU image (Enginuity 5773 - 5876) 285
Unmap devices 286
Unmap devices from EA or EF ports 286
I/O activity and unmapping devices 287
Introduce devices to a host 287
Introduce devices to Sun Solaris systems 288
Introduce devices to HP-UX systems 288
Introduce devices to IBM AIX systems 288
Introduce devices to HP Tru64 UNIX systems 288
Introduce devices to Windows systems 289
Configure dynamic SRDF-capable devices (Enginuity 5773) 289
Enable dynamic SRDF on the device (Enginuity 5773) 290
Set SRDF group attributes 291
Swap RA groups 291
Convert directors (Enginuity 5876 only) 291
Restrictions when converting directors (Enginuity 5876 only) 292

Port to director emulation support (HYPERMAX OS 5977) 292
Associate ports to director emulations (HYPERMAX OS 5977) 293
Disassociate ports from director emulations (HYPERMAX OS 5977) 294
Set port characteristics 296
SCSI protocol flags 297
Fibre protocol flags 299
Setting a port to show ACLX device 300
Report flag details 301

Chapter 8 Managing multiple iSCSI targets 303

Manage multiple ISCSI targets 304
Create IP interface on SE director port 305
Modify IP interface on SE director emulation 306
Delete IP interface on SE director emulation 308
Create iSCSI target on SE director emulation 309
Modify iSCSI target 311
Delete iSCSI target 313
Rename iSCSI target 314
Attach IP interface to iSCSI target 315
Detach IP interface from iSCSI target 316
Add IP route to SE director emulation 318
Remove IP route from SE director emulation 319
Example iSCSI configuration 320
Symconfigure command restrictions for iSCSI targets 320
Set online or offline state for iSCSI targets 321
Report iSCSI target information 321
List array IP addresses 321
List array iSCSI targets 322
List array IP routes 323
Expanded symcfg and sympd reporting for iSCSI targets 324
Expanded port group management for iSCSI targets 324
Report iSCSI target port information 326
Show port group with iSCSI targets 326
Show masking view with iSCSI targets 326
List HBAs with iSCSI targets (HYPERMAX OS 5977 or higher) 327
List device information with iSCSI targets 327
List device assignments with iSCSI targets 328
List CHAP information with iSCSI targets (HYPERMAX OS 5977 or higher) 328
List login information with iSCSI targets 329
List storage group demand with iSCSI targets 329

Chapter 9 Configuring Virtual Provisioning 331

Virtual Provisioning overview 332
Virtual Provisioning overview (HYPERMAX OS 5977) 332
Virtual Provisioning overview (Enginuity 5773 through 5876) 333
Create DATA devices (Enginuity 5773 through 5876) 334
DATA device configuration rules 335
Create thin devices (Enginuity 5773 through 5876) 335
Create thin devices using a file 335
Thin device configuration rules 337
Create thin pools (Enginuity 5773 through 5876) 337
Thin pool restrictions and rules 339

Rename thin pools 339
Set pool attributes (Enginuity 5773 through 5876) 339
Perform actions on thin devices (Enginuity 5773 through 5876) 341
Bind thin devices 342
Unbind thin devices 342
Rebind thin devices 343
Verify device pool binding 344
Allocate, reclaim and free space on thin devices 344
Allocate space on thin devices 345
Reclaim allocations on thin devices 345
Free written tracks on thin devices (Enginuity 5874 through 5876) 346
Free all allocations or written tracks (HYPERMAX OS 5977 only) 347
Background operations on thin devices 347
Stop background operations on thin devices 348
Drain and rebalance a thin pool (Enginuity 5874 through 5876) 348
Thin pool draining rules and restrictions 348
Deactivate devices in a thin pool 349
Start the drain on DATA devices 350
Activate devices in a thin pool 350
Stop the drain on DATA devices 350
Start pool rebalancing 351
Stop thin pool rebalancing 352
Automated pool rebalancing 353
Virtual Pool (VP) compression (Enginuity 5876 only) 353
Enable or disable thin pool VP compression 353
Start or stop VP compression on thin devices 354
VP compression rules 354
Verify pool and device states 355
Monitor thin pools 355
View thin pools 356
View thin device details 356
View all thin device pools 357
View thin device allocations bound to a different pool 358

Chapter 10 Federated Tiered Storage 361

Federated Tiered Storage (FTS) and FAST.X 362
FAST.X overview for HYPERMAX OS 5977 362
FTS overview for Enginuity 5876 363
Solutions Enabler features supported in FAST.X and FTS 363
eDisks 364
eDisk encapsulation 364
eDisk external provisioning 365
Geometry of eDisk encapsulated devices (Enginuity 5876) 365
eDisk Geometry limited devices restrictions (Enginuity 5876) 365
Create external disk group 366
Delete an external disk group 367
Add an eDisk 368
Rules and restrictions for adding eDisks 368
Show external disk information 369
Remove an eDisk 370
List external spindle state (HYPERMAX OS 5977 or higher) 371
Verify eDisk state (HYPERMAX OS 5977 or higher) 371
Start drain on eDisk (HYPERMAX OS 5977 or higher) 372
Stop drain on eDisk (HYPERMAX OS 5977 or higher) 373
Activate eDisk (HYPERMAX OS 5977 or higher) 374

Chapter 11 Device masking with Auto-Provisioning Groups 377

Auto-provisioning groups ... 378
Provisioning limits ... 378
Create a masking view ... 379
Auto-provisioning session rollback ... 380
Create an auto-provisioning session ... 380
Discover host HBAs ... 381
List host HBAs ... 381
Create groups and views ... 382
Manage masking views ... 385
Delete masking views ... 385
Name groups and masking views ... 386
Back up and restore masking views ... 386
Copy groups and masking views ... 387
Manage storage groups ... 387
Add devices to storage group ... 388
Remove devices from storage group ... 389
Rename a storage group ... 389
Delete a storage group ... 389
List and show storage group information ... 390
Manage port groups ... 390
Add ports to port group ... 390
Remove ports from port group ... 391
Delete port groups ... 391
Copy a port group from array to array ... 392
List and show port group information ... 392
Fibre Channel ID lockdown ... 392
Locking down a Fibre Channel ID ... 392
Manage initiator groups ... 393
Add initiators to initiator group ... 393
Remove initiators from initiator group ... 394
Deleting initiator groups ... 394
Initiator group flags ... 395
Set HBA flags ... 396
Replacing a HBA ... 398
Rename a HBA ... 399
Using CHAP authentication ... 399
Verify the auto-provisioning database ... 400
Display auto-provisioning group information ... 401
Modified reporting commands for HYPERMAX OS 5977 ... 401
Display masking views ... 402
View auto-provisioning group details ... 403
View auto-provisioning device assignments ... 405
Display device list with no auto-provisioning assignments ... 405
List initiator group devices ... 406
View the HBA alias name ... 406

Chapter 12 Masking Devices (Enginuity 5773 or lower) 409

Device Masking overview ... 410
VCMDB ... 410
Host access process ... 410
Access management for unsupported host platforms ... 411
Device masking controls ... 411
Device database locking ... 411

Device database lock release 411
Supported device masking topologies 412
HBA initiator support 412
Initial device masking configuration 412
Access control environment setup for device masking 413
Identify components for device masking 413
List physical device names of device-masking devices 414
List HBAs and FA director ports 414
List host connections to DMX director ports 414
List devices mapped to FA director 415
Initialize and update the VCMDB database 415
Enable iSCSI authentication 415
iSCSI CHAP authentication: Enginuity version 57xx 415
iSCSI software driver configuration 419
Configuring iSCSI with CHAP authentication 420
Configuring iSCSI for two-way chap authentication 423
Configuring iSCSI without CHAP authentication 423
RADIUS server configuration 425
Recommended actions for activating the VCMDB configuration 426
Discovering host HBAs 427
HBA identifiers and alias names 427
HBA identifiers and ASCII format 428
Rename HBA identifiers 428
Add masked devices to HBA/director-port channel 428
Add masked devices 428
Add masked reserved devices 429
Add masked device group devices 429
Add masked devices using a device file 430
Add devices with dynamic LUN addressing 430
Remove masked devices 431
Remove masked metadevice members 432
Remove masked devices from device group 432
Remove devices using a device file 433
Device masking database maintenance 433
View masking database 433
Initialize and clear masking database 433
Sync masking data 434
Unauthorized modification of the VCMDB 434
Create VCMDB backup file 435
VCMDB database types 435
Convert a VCMDB type 435
Restore a backup VCMDB 436
Restore a backup and converting its type 436
Block direct writes to the VCMDB 436
View the VCMDB 437
View contents of VCMDB 437
View VCMDB records for a director 437
View masked assignment to specific HBA 437
View device capacity 437
View HBA assignments 438
View the login history table 438
Refresh the database 439
Manage HBA initiators 440
List AWWN logins 440
Rename AWWNs 440
Replacing an HBA 441

Fibre Channel-to-host interface management 442
Find the FCID of a switch 442
Fibre Channel ID lockdown 443
FCID format 444
Device LUN visibility 444
LUN base/offset skip adjustment 445
Multiple hosts and broken sequences 445
HBA port flags 446
Heterogeneous host configuration 447

Chapter 13 Managing Quality of Service 451

Quality of Service overview 452
Copy pace and priority 452
Set copy pace and priority 453
Copy pacing examples 454
Device group QoS query 454
Copy pace settings output display limitations 455
Host I/O priority (Enginuity 5876 or lower) 455
Enable and disable priority of service 456
Set host I/O priority 456
List priority of service 456
Set bandwidth for Open Replicator (Enginuity 5876 or lower) 457
Set and reset LRU parameters 457
RDF director workload distribution 458
Enable and disable the RDF director workload distribution 459
Set the RDF director workload distribution 459
Reset the RDF director workload distribution 459
Workload based redistribution of RA QoS values 459
Display RDF workload distribution 460
View the audit log 460
Dynamic cache partitions (Enginuity 5876 or lower) 460
Create cache partitions 461
Enable and disable cache partitions 461
Device addition to a cache partition 462
Add devices to a cache partition 462
Modify a cache partition 463
Rename a cache partition 463
Delete a cache partition 464
Remove devices from a cache partition 464
Analyze mode for cache partitions 464
Set and display SRDF/A output based on cache partition values 465
Preserve empty cache partitions 465
List cache partitions 466

Chapter 14 Virtual LUN Technology (Enginuity 5874 through 5876) 469

Virtual LUN Technology overview 470
Migration requirements by feature 470
Virtual LUN technology restrictions 471
Virtual LUN VP mobility 471
Device pool-level migrations 472
VP compression guidelines for migration 472
FTS support for VLUN migration 472
Create virtual LUN migration sessions 472
Specify source devices for VLUN migration 473

Specify target criteria 474
Establish a migration session 474
Validate a migration session 474
Migration to configured space 475
Migrate to configured space using a device file 475
Migrate to configured space specifying target criteria 475
Error reporting for VLUN migration (configured space) 476
Migration to unconfigured space 477
Error reporting for VLUN migration (unconfigured space) 478
Migrate to a thin pool 478
View a thin migration 479
Virtual LUN migration management 480
List VLUN migration sessions 480
Query VLUN migration sessions 481
Verify VLUN migration sessions 483
Terminate VLUN migration sessions 483
View the audit log 484

Part 2 Querying and Reporting 485

Chapter 15 Configuration Query Operations 487

Configuration data overview 488
SCSI-level data 488
Array-level data 492
Device-level data 519
Disk-level data 545

Chapter 16 Statistics 555

Statistics overview 556
Object reporting statistics 556
Display options for performance statistics 557
REQUESTS statistics 557
List array statistics 558
List device statistics 558
List SRDF/A session statistics 558
List R-Copy session statistics 559
BACKEND statistics and PREFETCH activity 559
List BACKEND statistics 559
List PREFETCH activity 559
Director statistics 560
List director I/O requests 560
R-Copy session director statistics 560
PORT statistics 561
List director port statistics 561
List rcopy session director port statistics 561
List ISCSI statistics (GigE) 561
List MEMIO statistics 562
Disk statistics 563
List statistics for a specific disk 563
List spindle statistics 563
CACHE statistics 564
List front-end director cache statistics 564
List SRDF/A session cache statistics 564

List PATH statistics 566
List CYCLE statistics 566
RDF statistics 568
List SRDF/A session statistics by RA group 568
List SRDF device-level statistics 568
List SRDF statistics by director number 569
SRDF link-level statistics (Enginuity 5876 or lower) 569

Chapter 17 Change Tracker (Enginuity 5876 or lower) 571

Change Tracker overview 572
Practical uses for Change Tracker 572
Change Tracker command summary 573
Monitor data objects for changes 574
Create a Change Tracker session 575
Set Change Tracker session type 575
Create a Change Tracker device text file 575
Mark a data object for change monitoring 576
List objects marked for change monitoring 576
Remove mark from change monitoring database (without termination) 577
Remove marked objects from change monitoring database (with termination) 577
View objects marked for change monitoring 578
Change tracker reports 579
View change tracker reports 579
Example procedure: Creating a report of change tracking information 581
Change Tracker for SRDF capacity planning 582
Using Change Tracker to perform SRDF capacity planning 583

Chapter 18 Optimizing Array Performance (Enginuity 5876 or lower) 585

Optimizer overview 586
Optimizer configuration requirements 587
Optimizer operations 588
Optimizer performance metrics 588
Optimizer modeling service time 589
Optimizer device swap determination 589
Optimizer swap determination 589
Optimizer swap procedural overview (Enginuity 5874 to 5876) 590
Optimizer swap procedural overview (Enginuity 5773 or lower) 591
Optimizer operations using the symoptmz command 592
Enable and disable Optimizer 593
Commit command file changes 593
Clear Optimizer statistics 594
Set control parameters 594
Set advanced parameters 595
Optimizer time windows 595
Set the swap priority for devices 595
Set manual swap lists for Optimizer (Enginuity 5874 and 5876) 596
Set manual swap lists for Optimizer (Enginuity 5773 and lower) 596
Reviewing Optimizer plans 596
Approve a swap list 602
Perform Optimizer rollbacks 602
Read Optimizer logs 602

Optimizer device migration (Enginuity 5773 or lower) 603

Chapter 19 Managing Time Windows (Enginuity 5876 or lower) 605

Time windows overview 606
Time window types 606
Time window definitions 606
Time window restrictions and guidelines 607
Time window weekly format 607
Time windows conversion 607
Time Windows Conversion guidelines 607
Time Windows conversion examples 608
Add time windows 609
Remove time windows 610
Remove all time windows 612
Display time windows 612
List time window definition summary 613

Chapter 20 Managing Network IPsec (Enginuity 5876 or lower) 615

Network IPsec Overview 616
IPsec and EMC arrays 616
SYMCLI IPsec management component 617
IPsec key management 617
View policy information 618
List IPsec policies 618
Show IPsec policy detail 618
List IPsec policy statistics 620
Command file scripting for creating/modifying policies 621
Command file syntax adding/modifying policies 622
Script IPsec policy syntax 622
Script proposal set syntax 624
Commit a policy-scripted command file 627
Preview the IPSEC command file 627
Commit IPSEC command file 628
Delete a policy-scripted command file 628

Chapter 21 Events and Logs 629

Events and logs overview 630
Log option configuration 630
VMAX event monitoring using SYMCLI 632
Monitoring events 632
List events 632
Common audit log 633
Show audit log for single array 633
List audit log for specified time period 634
Event daemon 635

Chapter 22 XML Structured Output 637

XML structured output overview 638
XSLT: XML data transformations 638
Element-based XML 638
Set XML mode with SYMCLI 639

XML output using SYMCLI 639

Part 3 Migrating Data 641

Chapter 23 Open Replicator Operations 643

Open Replicator overview 644
ORS practical uses 644
ORS limitations for HYPERMAX OS 5977 or higher 644
ORS and host interaction 645
ORS rcopy concepts 645
ORS operational rules and limitations 647
ORS device guidelines 649
ORS SAN setup requirements 650
ORS SYMCLI symsan support 651
Open Replicator command summary 651
Open Replicator session options 652
ORS hot pull copy session example 653
ORS cold push copy session example 654
Open Replicator control operations 655
ORS and creating a device file 656
Create ORS copy session 658
Activate ORS session 665
Monitor ORS session status 668
Terminate an ORS session 674
Remove an ORS remote device from a session 675
Recreate an ORS session 675
Recover from failed ORS session 676
Restoring an ORS session 677
VDEVs in Open Replicator 678
Open Replicator examples 680
Example 1: Perform an ORS hot pull operation 680
Example 2: Pushing data using BCVs as the control devices 685
Example 3: Pull data online from IBM F20 to VMAX array 691
Example 4: Pushing online data from VMAX array to HDS 9960 array 698
Example 5: Perform a cold push from a VDEV 704

Chapter 24 Operational Restrictions and State Reference 709

Device control operations allowed for ORS control devices 710
ORS operations allowed for replication session states 710
TimeFinder/Mirror and TimeFinder/Clone sessions 711
Snap VX sessions 717
SRDF sessions 720
ORS operations allowed for device types 723

Chapter 25 Federated Live Migration Operations (Enginuity 5876 or lower) 725

Federated Live Migration overview 726
The FLM process 727
FLM array/device requirements 727
FLM array/device limitations 728
Federated Tiered Storage support for FLM 728
Configure the application hosts for FLM 728

FLM prerequisites for zoning and masking 729
Migration using FLM 729
Create a device file for FLM 730
Create an FLM session 730
Verify the created FLM session 731
FLM session activation 732
Terminate FLM session 737
FLM – restore new array device identity 738
Restore donor device to host-active mode 738

Index **741**

FIGURES

1 Access Control Lists and Entries in the Access Control database 60
2 Example default configuration 77
3 Establish Administrator 79
4 Grant all permissions to non-pooled devices 79
5 Host-visible GNS state 150
6 GNS across multiple hosts 151
7 Concatenated metadevices 252
8 Striped data 253
9 High-level overview of FAST.X environment 362
10 Masking view overview 380
11 Masking view MV_1 384
12 Device masking components to identify 414
13 A DeltaMark bitmap table 572
14 SRDF configuration 583
15 Optimizer application architecture 586
16 Swap steps using one DRV 590
17 Identify volumes to swap 591
18 Copy volume to DRV 591
19 Copy DRVs to new locations 592
20 DRVs are split 592
21 Control array device pull operation 646
22 Control array device push operation 647
23 Control array hot pull using the symrcopy command 654
24 Control array device cold push using the symrcopy command 655
25 Cold push copy session: single device to multiple remote devices 662
26 Example FLM configuration 727

TABLES

1 Typographical conventions used in this content 25
2 SYMCLI help options 30
3 Abbreviated command line actions 33
4 User authorization command file format 54
5 Initial access groups set 63
6 ACL setup for new array installation 63
7 Access Control Permissions: AccessType 70
8 Logical name formats 134
9 Location of daemon authorization file 154
10 Restarting the GNS daemon 155
11 Locations of GNS configuration options 157
12 FAST version comparisons (Enginuity 5874 through 5876) 181
13 Differences between FAST and FAST VP 202
14 FAST State values 205
15 Current Activities values 206
16 Fast control parameters for symfast CLI 207
17 Group Attribute values 213
18 Group State values 213
19 Plan State values 213
20 Compliance Report output values 217
21 Technology Demand Report output values 220
22 Configuration change operations 230
23 Possible metric names (HYPERMAX OS 5977) 234
24 Possible metric names (Enginuity 5773 through 5876) 234
25 Maximum device sizes (HYPERMAX OS 5977) 237
26 Steps to create a device 240
27 Maximum device sizes (Enginuity 5773 through 5876) 243
28 Number of data members for RAID 5 and RAID 6 devices 244
29 Local and remote device assignments 245
30 Metadevice sizes by Enginuity version for auto_meta 252
31 Steps to create an SRDF BCV metadevice 253
32 Metadevice data preservation 258
33 Thin device conversions 263
34 Values for Effective Device Geometry 270
35 SCSI protocol port flags 297
36 Fibre protocol port flags 299
37 VP operations for thin devices 332
38 VP operations for thin devices 333
39 Supported thin metadevice features 335
40 Maximum subscription percent capacities 338
41 Drain and rebalance command for thin pools 348
42 Solutions Enabler features supported in FAST.X and FTS 363
43 Provisioning limits for arrays running HYPERMAX OS 5977 or higher 378
44 Provisioning limits for VMAX arrays running Enginuity 5874 through 5876 379
45 Initiator group flags 395
46 Supported HBA flags 397
47 LUN base/offset scenarios for multiple hosts with skip holes 446
48 HBA port flags 447
49 Host platforms and interface configuration flags 448
50 Virtual LUN requirements by feature 470
51 Audit log session information 484

52 Pool Rebalancing Parameters 518
53 Device types - HYPERMAX OS 5977 520
54 Device types - Enginuity 5876 and lower 520
55 Change Tracker command summary 573
56 Log file configuration options 630
57 Control and remote device guidelines 649
58 Device control operations allowed for ORS control device states with host 710
59 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Mirror SOURCE 711
60 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Mirror TARGET 712
61 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror SOURCE 712
62 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror TARGET 713
63 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Clone SOURCE 714
64 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Clone TARGET 715
65 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone TARGET 715
66 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone SOURCE 716
67 Allowable rcopy operations when the control device for PUSH session is in use as a Snap VX SOURCE 717
68 Allowable rcopy operations when the control device for PUSH session is in use as Snap VX TARGET 718
69 Allowable rcopy operations when the control device for PULL session is in use as a Snap VX SOURCE 719
70 Allowable rcopy operations when the control device for PULL session is in use as a Snap VX TARGET 719
71 Allowable rcopy operations when the control device for PUSH session is in use as an SRDF R1 mirror 720
72 Allowable rcopy operations when the control device for PUSH session is in use as an SRDF R2 mirror 721
73 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R1 mirror 721
74 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R2 mirror 722
75 ORS operations allowed by device type 723

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 EMC Online Support (https://support.emc.com). Check 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 on page 27 describes how to use the SYMCLI configuration commands to set up your array devices.
- Part 2: Querying and Reporting on page 485 describes how to use the SYMCLI query commands and various reporting tools to monitor and manage your storage array performance.
- Part 3: Migrating Data on page 641 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 Solutions Enabler information:

EMC Solutions Enabler Release Notes
Describes 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 the product guide, physical planning guide, and power documentation for VMAX3 arrays.

EMC VMAX Family (10K, 20K, 40K) Documentation Set
Contains documentation related to the VMAX 10K, 20K, and 40K arrays.

EMC VMAX3 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. After it is installed, familiarize yourself with the information under Help tab.

Special notice conventions used in this document
EMC uses the following conventions for special notices:

DANGER

Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

Addresses practices not related to personal injury.

Note

Presents information that is important, but not hazard-related.

Typographical conventions
EMC uses the following type style conventions in this document:

Table 1 Typographical conventions used in this content

Bold	Used for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
Italic	Used for full titles of publications referenced in text
`Monospace`	Used for: • System code • System output, such as an error message or script • Pathnames, filenames, prompts, and syntax • Commands and options
`Monospace italic`	Used for variables
`Monospace bold`	Used for user input
[ ]	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 as follows:

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 information about EMC products, go to EMC Online Support 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 help us improve the accuracy, organization, and overall quality of the documentation. Send your comments and feedback to:
VMAXContentFeedback@emc.com

PART 1

Setting up and Configuring Storage Arrays

This section describes how to set up and configure storage array devices using the Solutions Enabler SYMCLI commands.

Chapters include:

Chapter 1, "Introduction "

Chapter 2, "Discovery"

Chapter 3, "User and Group Authorization"

Chapter 4, "Host-based Access Control"

Chapter 5, "Grouping Devices"

Chapter 6, "Fully Automated Storage Tiering"

Chapter 7, "Manage Configuration Changes"

Chapter 8, "Managing multiple iSCSI targets"

Chapter 9, "Configuring Virtual Provisioning"

Chapter 10, "Federated Tiered Storage"

Chapter 11, " Device masking with Auto-Provisioning Groups"

Chapter 12, "Masking Devices (Enginuity 5773 or lower)"

Chapter 13, "Managing Quality of Service"

Chapter 14, "Virtual LUN Technology (Enginuity 5874 through 5876)"

CHAPTER 1

Introduction

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

- Solutions Enabler overview 30

Solutions Enabler overview

The EMC® Solutions Enabler SYMCLI provides management capabilities for storage environments. The SYMCLI is a specialized library consisting of commands run at the command line, or used within scripts. These commands monitor device configuration and status, and perform control operations on devices and data objects within a VMAX-based storage environment.

The SYMCLI resides on a host system to monitor and perform control operations on DMX and VMAX Family storage arrays. SYMCLI commands are run 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).

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 replication operations.

SYMCLI help options

SYMCLI provides the following multiple options to obtain top-level help using the command set:

Table 2 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

Syntax

To display command line help for a specific command, use the following syntax:

```
command -h
```

Examples

To display help for the **symcfg** command, enter:

```
symcfg - h
```

Each command has its own man page for command line reference. To view the man page for the **symcfg** command, enter:

```
man symcfg
```

In a UNIX environment, the SYMCLI man page directory (/usr/symcli/man/) must be included in your MANPATH environment variable.

In a Windows environment, the default directory for man pages is C:\Program Files\EMC \symcli\man.

Device and object references

There are a number of different terms used with SYMCLI commands to identify and specify a VMAX array device:

- *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

Environment variables are set to streamline the command line session. Setting environment variables to commonly used values, eliminates the need to specify those arguments on the command line.

Display environment variables

Examples

To view a list of environment variables that can be set for the SYMCLI session, enter:

```
symcli -env
```

To view the current environment variables, enter:

```
symcli -def
```

Enable environment variables

Syntax

From UNIX host, set an environment variable, use the following syntax:

```
setenv variable_namevalue
```

From Windows host, set an environment variable, use the following syntax:

```
set <environment variable>=<value>
```

Examples

For UNIX host, to enable the `SYMCLI_VERBOSE` variable to set the default command behavior to always display command output with details:

```
setenv SYMCLI_VERBOSE 1
```

For Windows host, to enable the SYMCLI_VERBOSE variable to set the default command behavior to always display command output with details:

```
set SYMCLI_VERBOSE=1
```

Disable environment variables

Syntax
From UNIX host, turn off an environment variable, use the following syntax:

```
unsetenv variable_name
```

From Windows host, turn off an environment variable, use the following syntax:

```
unset <environment variable=
```

Examples
From UNIX host, to turn off the verbose mode, enter:

```
unsetenv SYMCLI_VERBOSE
```

From Windows host, to turn off the verbose mode, enter:

```
set SYMCLI_VERBOSE=
```

Display full group names

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

Syntax
To display the full group name in the list command output, use the following syntax:

```
setenv SYMCLI_FULL_NAME sid
```

Example
To display full group name for array 567, enter:

```
setenv SYMCLI_FULL_NAME 100200000567
```

Preset names and IDs

Description
To reduce repeated key strokes for a group of commands that require the same argument, set the SYMCLI_DG and SYMCLI SID to a preset value.

Syntax
To preset the device group name, use the following syntax:

```
setenv SYMCLI_DG dgName
```

To preset the array ID, use the following syntax:

```
setenv SYMCLI_SID array_id
```

Examples
To preset the array ID `100200000567` for a series of consecutive commands, enter:

```
setenv SYMCLI_SID 100200000567
```

To preset the device name for all `-g` arguments, enter:

```
setevn SYMCLI_DG MyDG
```

Command input and output tips

There are a number of different terms and abbreviations used with SYMCLI commands to identify and specify a VMAX array device:

Command line time-saving tips

Actions can be abbreviated, as follows:

Table 3 Abbreviated command line actions

Command	Abbreviated command
`symcfg discover`	`symcfg dis`
`symcfg -sid 000002304324 sync`	`symcfg -sid 24 sync`
`symcfg LIST`	`symcfg list`
`sympd show /dev/rdsk/c2t1d1s2`	`sympd show c2t1d1s2`

Truncated output display

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

Version compatibility information and restrictions

Compatibility between Solutions Enabler versions, can be set for output displays, running older version scripts, and client/server interoperability.

Set compatibility mode for older Solutions Enabler versions

Description
Compatibility mode is set globally using the `setenv SYMCLI_MODE` environment variable, 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 V75, V76 and V80.

Examples
For example, to set the compatibility mode to Solutions Enabler V7.5, enter:

```
symcfg list -mode V75
```

Attempts to specify a compatibility mode earlier than V7.5 fails with an Invalid mode error.

```
symauth -sid 111 -mode V74
'V74': Invalid mode specified.
```

Prior version Solutions Enabler scripts

Specialized scripts developed using a prior version of Solutions Enabler can be run 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, output that is only available in later releases may not be returned.

CLI displays with older Solutions Enabler versions

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 are 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 version compatibility

Solutions Enabler V8.1 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 V8.1 are rejected.
- SYMAPI server connections from clients running Solutions Enabler earlier than V7.5 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.

- Attempts to open a database file written by a version prior to V7.5 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.
```

- When comparing client and server versions, only the major and minor versions are compared (for example: 7.4, 7.5). The edit and patch levels are ignored.

CHAPTER 2

Discovery

This chapter describes the SYMCLI discovery process.

- Discovery overview 38
- Discovery syntax options 38
- Verify host configuration database is synchronized 38
- Scan for new devices 39
- Connectivity authorization 39
- Configuring virtual disk mapping 40
- Display virtual disks as array devices. 41
- Host configuration database file 41
- Configuration data synchronization 43

Discovery overview

The discovery process retrieves array, volume-level configuration, and status information. Discovery refers to the process where 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 the environment is discovered, information requests are made to retrieve array-level (high-level) data or device-level (low-level) information from it using SYMCLI commands.

Before using the SYMCLI commands, run the `symcfg discover` command to build the configuration (SYMAPI) database. Do this once after installation, and after any changes are made to the array configuration.

Discovery syntax options

Description

During the 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 the host. The `discover` 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. For more information about the configuration database, refer to Host configuration database file on page 41

Syntax

To run a discovery operation, use the following syntax:

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

Options

-pdev
Limits the discovery operation to only physical device information.

-sid
Limits the discovery operation to a specific array.

Examples

To scan the hardware and rebuild the database, enter:

```
symcfg discover
```

Verify host configuration database is synchronized

Description

The `symcfg verify` command verifies if the host configuration database file is synchronized with the configuration of an array. If they are in sync, the verify action returns code `0` (the `CLI_C_SUCCESS` value). If they are out of sync, the verify action returns code `24` (`CLI_C_NOT_IN_SYNC` value).

The `symcfg list -status` command lists all arrays connected to the host, and shows whether the configuration has changed.

Examples

To run the verify action for a specific array, enter:

```
symcfg verify -sid 814
```

To run the list action, enter:

```
symcfg list -status
```

Sample output

For the `symcfg verify` command:

```
The Symmetrix configuration and the database file are in sync.
echo $status 0
```

For the `symcfg list -status` command:

```
                          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
```

Scan for new devices

Description

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. The `scan` operation searches the 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. If changes are associated with array devices, follow the `scan` operation with the `discover` operation.

Note

The `symcfg scan` command is only supported on UNIX platforms.

Examples

To scan for new devices, enter:

```
symcfg scan
```

To discover changes associated with array devices, enter:

```
symcfg discover -symmetrix
```

Connectivity authorization

Description

Some arrays may require authorization information to provide access to the array. Use the `symcfg authorization` command to supply this information for use in

subsequent discovery operations. The `symcfg authorization` command lists, adds, updates, or deletes this connectivity information. The `update` action updates the password of an existing entry.

Syntax

To authorize connectivity, 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, configure the host access credentials and copy the SSL certificate from the server to the VM OS.

Procedure

1. Configure host credential using the following syntax:

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

2. Copy the SSL certificate. For example from the server: `/etc/vmware/ssl/rui.crt`) to the VM OS as: `/var/symapi/config/viclient_cert.pem`
3. When virtual disk mapping is configured, use following syntax to return the virtual disk information:

   ```
   ./syminq
   ```

Results

```
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` commands. 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 list` output shows. In some cases, `syminq` shows larger capacity than `sympd list`, because the VMware datastore (or virtual disk) can have multiple array devices.

Display virtual disks as array devices.

Example

To display virtual disks as array devices for array `266`, enter:

```
sympd list -sid 266
```

Sample output

```
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
```

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`

An additional configuration database (**.bin**) file can be created to meet specific requirements.

Note

In a multi-database environment, verify that the correct database is being used. Always confirm the current database before applying a command. For safe command line environments, it is recommended to 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 name

Procedure

1. Check which host configuration database file is being used:

   ```
   symcli -def
   ```

2. 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
   ```

Database location 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

SYMCLI commands are run 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

Example

To view basic configuration information about the current database in use, enter:

```
symcfg -db
```

Sample output

```
symcfg list -db
  Type of SYMAPI Database                          : Full
  Host Node Name which discovered the DB           : api33
  Host OS Type which discovered the DB             : LINUX
  Version of SYMAPI Library which discovered the DB : 8.0.X.X (Edit
Level: 2001)
  Version of SYMAPI Library which wrote the DB     : 8.0.X.X (Edit
Level: 2001)
  Min Edit Level of SYMAPI Lib Reqd to Read the DB  : 2000
  Time Database Was Last Synchronized              : Mon Feb  9
14:33:20 2015
  Time Any Device Group Was Last Modified          : N/A
```

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.

Configuration data synchronization

A `sync` operation of the host configuration database with the storage environment configuration, performed once a discovery operation has occurred, 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.

Update configuration information (sync operation)

Description

The `sync` operation only updates state, mode, and configuration information (if a configuration change has occurred). It does not update statistical information. Statistics on page 555 provides information about obtaining array performance statistics. The `symcfg sync` command 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 are not added to the database during the synchronization. Synchronizing data is an array-specific operation.

Syntax

To perform a `sync` operation, use the following syntax:

```
symcfg sync
```

Options

-sid SymmID
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

Inhibit Database synchronization

Description

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

Examples

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
```

It may be necessary to refresh the database if executing commands while offline mode is enabled or execute commands that normally run in the offline mode, such as the `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.

- User and group authorization overview 48
- Authorization settings management 51
- Monitor authorization 56

User and group authorization overview

User authorization restricts 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 59 for more information on the Access Control database.

The `symauth`command maps a user or group to a specific role, and allows the user or group access to specific operations 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 determines the operations a user can perform. Unlike host-based access control, a user is assigned a particular role for the entire array or specified components within the array. For each array, a user or group can be assigned up to four roles. Roles are predefined and cannot be changed.

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 User ID formats on page 50 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). The StorageAdmin also has 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.

List roles supported by the array

Syntax

To list the roles supported by an array, enter:

```
symauth list -roles
```

Assign multiple user roles

Examples

The `symauth` command is enhanced to assign multiple roles (up to four) separated with a '+' character.

```
StorageAdmin+Auditor+Monitor
```

Note

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

The`remove` command removes roles only for users assigned multiple roles, as shown in this 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 lower supports only one role per user or group. When a user has multiple roles defined on an array running Solutions Enabler 8.0 or higher, older versions of Solutions Enabler show only the highest assigned role for that user.

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:

1. Verifies if user has access to the entire array.
2. Checks if group has access to the entire array.
3. Verifies if user has access to the ThinPool or StorageGroup component within the array.
4. Checks if group has access to the ThinPool or StorageGroup component within the array.

The rights granted by each check are combined together to form an overall set of rights for the user or group. For example, if a user is assigned to the SecurityAdmin role and also belongs to finance group, that is assigned the Monitor role, then the user 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.

User authentication types

Options

L

Indicates a user or group authenticated by LDAP. 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 or group authenticated by the SMC server. For example:
`C:Boston\Legal`— Indicates that user group Legal logged in through SMC sever Boston.

Examples

Only Virtualization domain users or groups are allowed the role of StorageAdmin, with access to specific array components such storage groups or thin data pools. To assign thin pool access to a Virtualization domain use, enter:

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

Username and groupname formats

Description
Within role definitions, user IDs or group IDs are 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.

Examples

- `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.

Wildcards
When using a wildcard (such as the asterisk), 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, it is used.
- If a partial match (for example, `D:*\putman`) is found, it is used.
- If an unqualified match (for example, `putman`) is found, it is used; otherwise the user role is 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 invalid through API validity checks.

Authorization settings management

The `symauth` command manages user and group authorization. This command enables and disables user and group authorization, sets enforcement mode, lists defined users and groups with their roles, and assigns roles to users and groups.

Current username identification

Options

-username
Displays the current username.

Examples
To display all groups to which the user belongs, enter:

```
symauth show -username
```

Sample output
This output 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.

User authorization

User and group role mappings should be defined before enabling authorization. At a minimum, there must be one mapping for an individual to the Admin or Security Admin, as user authorization can only be enabled if there is an individual (as shown by `symauth show -username` command) mapped to a role of either Admin or SecurityAdmin roles. These roles provide the ability to perform authorization control operations. Up to four roles can be assigned to a user group.

Set user authorization

Description
Assign user or group roles using the `symauth` command with role mappings supplied through a command file, or, on a UNIX host, supplied by 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 48 for supported roles), and end with a semicolon. For further details on using the command file, refer to User authorization command file usage on page 54.

Syntax
To create role mappings using a command file, use the following command syntax:

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

Examples

To create the user-to-role and group-to-role mappings, by supplying a redirection of STDIN (on UNIX host):

```
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;!
```

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.

Enable user authorization

Description
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 User ID formats on page 50 for assigning component specific authorization.

Only a user or group granted the role of Admin or SecurityAdmin can enable or disable authorization for an entire array.

Examples
To enable user authorization on array `097`, enter:

```
symauth -sid 097 enable
```

Note

Specifying an array ID is optional for `enable` action.

User authorization enforcement

Description
Once user authorization is enabled, failed access attempts are 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, is be set using the `symauth set enforcement` command.

Syntax
To set the security enforcement level, use the following syntax:

```
symauth -sid <SymmID> set enforcement <advise | enforce>
```

Options

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. .

Note

Enforcement can also be set using a command file as described in User authorization command file usage on page 54

.

Modify user authorization role mappings

Description

Modify user or group roles using the `symauth` command with role mappings supplied through a command file, or, on a UNIX host, supplied by 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 48 for supported roles), and end with a semicolon. For further details on using the command file, refer to User authorization command file usage on page 54.

Syntax

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

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

Options

-file

This is the fully qualified path of the file containing the user-to-role mappings.

Examples

To modify user or group role mappings supplying a redirection of STDIN (on Unix host), enter:

```
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 forThinPool 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.

User authorization command file usage

A command file can be used to set user authorization, so a sequence of operations are previewed and committed at once. A command file is a simple text file formatted as follows:

Table 4 User authorization command file format

Action	format
Add a user to roles	`assign user UserName to role RoleNames;`
Remove roles from user	`remove user UserName from role RoleNames;`
Reassign an existing group to roles:	`reassign group GroupName to role RoleNames;`

Table 4 User authorization command file format (continued)

Action	format
Delete a user	`delete user UserName;`
Set enforcement	`set enforcement [advise | enforce];`

Note

Assign operation is additive and does not replace an existing role. A user or group is allowed up to four authorization roles. Remove operation will remove individual roles for users with multiple roles.

Naming rules

- The following naming rules apply to user names and group names:
  - *UserName* or *GroupName* is the name of a user or a group (32-character maximum).
  - Spaces are allowed if the name is quote delimited.
  - Examples of a valid *Username*include: `"D:domain\joe", "H:host\joe", "domain\joe"`, and `"joe"`. For`"domain\joe"` it is interpreted as `"D:domain\joe"`. For `"joe"`, this *Username* is allowed regardless of domain or host.
- The following rules apply to *RoleNames*:
  - *RoleNames* are the roles assigned to or removed from a user or group.
  - Current valid roles include: *Admin, SecurityAdmin, StorageAdmin, Auditor, PerfMonitor, Monitor,* and *None*.
  - *RoleNames* are case insensitive.
  - To assign multiple roles to a user or group, place a plus sign "+" between roles (for example, **`assign user User1 to role StorageAdmin+Auditor`**).

Preview and commit user authorization actions

Description

To apply authorization actions (command file entries) to the array perform these operations on the command file:

1. Preview — This verifies the syntax and contents of the file.
2. Commit - This verifies and commits the contents of the command file to the array.

Syntax

To preview the command file entries, use the following syntax:

```
symauth -sid SymmID preview -file PathName
```

To commit the command file entries, use the following syntax:

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

Backup the authorization database

Description

The backup operation saves the contents of the user and group authorization database from the 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:

Syntax
To backup the database with a fully qualified path and filename for the backup file, use the following syntax:

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

Restore the authorization database

Description
The restore operation re-initializes the user and group authorization database on an array, from a previously generated backup file, and re-enables user authorization. The specified file must be created by an earlier backup operation and can be 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:

Syntax
To restore the database with the fully qualified path and filename for an existing backup file, use the following syntax and specify:

```
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. Otherwise, have a user with Admin or SecurityAdmin privileges re-enable user authorization on the array.

Monitor authorization

Description
Use the monitoring operations to view the current user authorization settings on a specific array.

Examples
To list the user authorization policies in effect on an array, enter:

```
symauth -sid 025 list
```

To list the supported roles, enter:

```
symauth list -roles
```

To list the component types, enter:

```
symauth list -components
```

List user roles, names, and components

Description
The `list` operation retrieves current user authorization information from the array. If current information cannot be retrieved, the operation fails.

Syntax

To list user authorization information, use the following syntax:

```
symauth [-sid SymmID] [-offline]
    list -users [-by_domain | -by_role | -by_user]
```

Options

-offline
: Retrieves cached authorization data on disk and does not communicate with the specified array. If there is no cached data on disk, no data displays (appearing as if there is no authorization data on the array.

-v
: Use this option to request a multi-line display or when the length of names or components exceeds the 79-character limit.

Examples

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

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

Sample output

The output shows the list the mappings of roles to user and group names and their applicable components for array 1234.

```
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.
```

Note

- User `H:mars\Jones` has *StorageAdmin* role and which grants access to array 1234. This authorization rule is active as noted by the period (.) in the `Flags` column.
- 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. 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 its group members.

CHAPTER 4

Host-based Access Control

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

- Host-based access control overview 60
- Create or modify access control data 61
- Create and manage access groups 64
- Create and manage limited access to access pools 67
- Create and manage access control entries 69
- Obtain access control information 75
- Verify a locked access control session 76
- Release locked access control sessions 76
- Access control strategies 76
- Back up and restore the access control database 81

Host-based access control overview

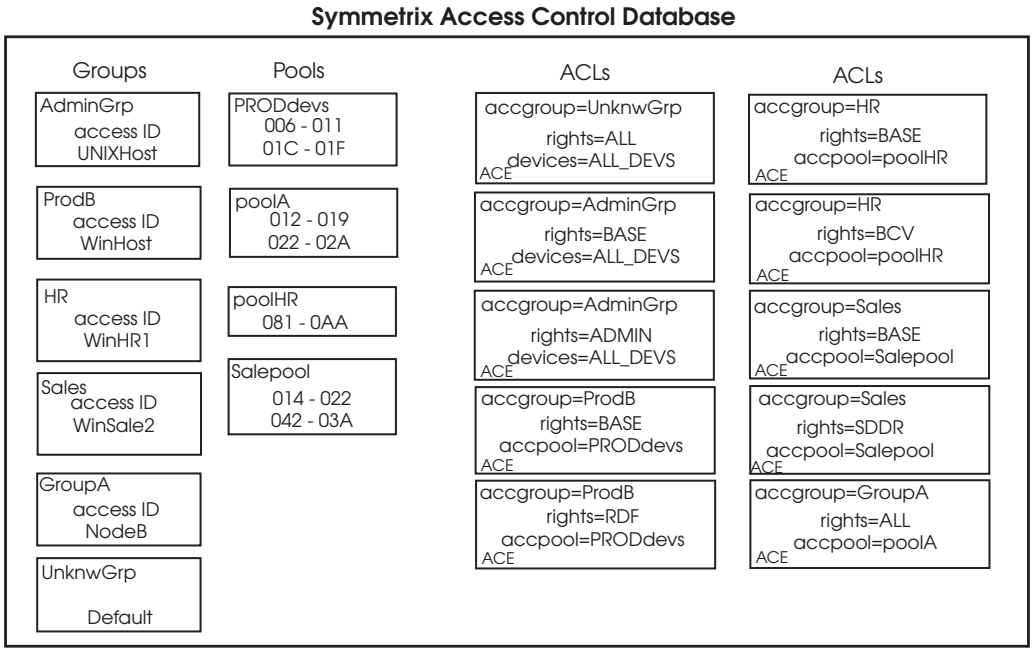
The **symacl** access control command sets up and limits access to array resources (access pools). The command sets up access control mechanisms and changes access control entries through the Access Control database.

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

Access Control database

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

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



The access control database contains the following access control components:

- **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 the `symacl -unique` command.
- **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 7 on page 70 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 VMAX Family 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. Other platforms use different characteristics of the host, such as a processor identifier.

Alternate access ID
An alternate access ID is randomly generated provided by a 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.

Forcible generation of alternate access ID

Description
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 error (`Unable to obtain unique ID for host`) 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 this, exit client/server mode on the client and execute the commands to generate a new access ID for the client.

Syntax
To forcibly generate a new alternate access ID using a supplied passphrase, use the following syntax:

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

To forcibly generate a new alternate access ID using a file that contains the passphrase, use the following syntax:

```
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.

Create or modify access control data

The `symacl` command creates or changes the access information in the Access Control database by committing a command file to the 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.

The `symacl` command performs the following operations (specified in the command file):

- Creates new access groups
- Adds and removes access IDs to access groups
- Moves an access ID from one group to another
- Creates new access pools
- Adds and removes devices to access pools
- Deletes access pools and access groups
- Adds ACEs to grant access
- Removes ACEs to deny access

Note

After Access Control changes are made, a discover operation is required (`symcfg discover`).

Applying access control operations

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

Procedure

1. `Preview`

   Verifies the syntax and accuracy of the contents of the entries in the command file.

2. `Prepare`

   Performs the preview checks, but also verifies the validity of the requested access control modifications against the current state of the access control database in the array.

3. `Commit`

   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 operations can ensure that the `commit` action is not rejected, or they can be used to debug the command file entries.

Minimum access control configuration

To initialize the Access Control Database for a new array installation, create a host-based administrator and an access pin. This is usually performed by an EMC Engineer and is described in the following example:

Table 5 Initial access groups set

Access groups	ID names
AdminGrp	SunHost1 ACCPIN
UnknwGrp	unknown

The name `SunHost1` shown in the table (associated with `AdminGrp`) is the ID name that designates the machine where access controls are administered. To perform `prepare, commit,` and `release` actions, an access ID (PIN) with the name **`ACCPIN`** is created in the delivered setup. This ID name is also used for the value setting of SYMCLI_ACCESS_PIN. For more information, see Create and manage access groups on page 64.

Initial ACL setup

As shown in the following table for the delivered setup described in Minimum access control configuration on page 62, access group `AdminGrp` has access to all devices (`ALL_DEVS`) and this group is 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. For initial 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 6 ACL setup for new array installation

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 6 on page 63.The VLOGIX privilege

returns as true since access is 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 a check for the VLOGIX privilege will now fail. For the `UnknwGrp` to still have VLOGIX privilege, that privilege must be explicitly granted to the `UnknwGrp` and associated with `ALL_DEVS`.

Create and manage access groups

Various sets of users tend to use the same applications that utilize common Solutions Enabler features from a given host. System users typically share the same device resources and require access permissions to these shared devices. For shared devices, 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 60).

Access groups contain groups of access IDs and their ID names. Any access ID and name can only belong to one group and are paired together into the database. For ease of management, it is highly recommended to use 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 is used to create access control entries (ACEs).

Verify administrative authority

Syntax

Prior to making any access control changes, verify the host used for ACL setup has the administrative authority. To verify the host, use the following syntax:

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

Create an access group

Examples

To add access groups named HR and Sales in array `12345`, enter n the command file:

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

To commit access groups to the database using the file `addnewgroups.cmd`, enter:

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

At this point, a prompt may display for a 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

Description

The host access `Id` is case sensitive, allows alphanumeric characters, underscores, dashes and no spaces. For `Id` name length:

- Arrays running HYPERMAX OS 5977 or higher, allows 1 - 31 characters.
- Arrays running Enginuity 5876 or lower, allows 1 - 8 characters.

Syntax

To add a host access ID to an access group, use the following syntax in the command file:

```
add host accid Id name ldName to accgroup GroupName
```

Note

The host access ID is encrypted; so choose an *ldName* that is easy to remember.

Examples

To add `WinHost` with an access ID `73900158-06174491-16225515` to the `ProdB` group, enter in the command file:

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

Note

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

To commit the file (`addnewgroups.cmd`) and add the host access IDs to the specified groups, enter:

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

Note

To update the array configuration after Access Control changes are complete, run the `symcfg discover` command. changes

User access IDs (PINs) for AdminGrp

Description

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`, the `symacl` command prompts for this PIN.

The user access `Id` is case sensitive, allows alphanumeric characters, underscores, dashes and no spaces. For `Id` name length:

- Arrays running HYPERMAX OS 5977 or higher, allows 1 - 31 characters.
- Arrays running Enginuity 5876 or lower, allows 1 - 8 characters.

Syntax

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 host access ID is encrypted; so choose an *ldName* that is easy to remember.

Examples
To add the administrative user access ID (`1234PIN`) for `JOEPIN`, enter in the command file:

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

Note

User access IDs are set only for the `AdminGrp` access group. Setting an ID for other groups will not return an error and may not prompt for a PIN.

To commit the file `addnewgroups.cmd` and add the user access IDs to the specified groups, enter:

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

Edit and manage an access group

Once access groups are established, access IDs or ACEs can be removed from a group, IDs can be moved between groups, or groups can be deleted.

Note

To update the array configuration after Access Control changes are complete, run the `symcfg discover` command. changes

Remove an access ID from a group

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

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

Examples
To remove user `HRUser2` from the `HR` group, enter in the command file:

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

To commit the file (`removeaces.cmd`) and remove the specified user, enter:

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

Move an access ID to another group

Syntax
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
```

Examples
To move user `HRUser1` to `GroupA`, enter In the command file:

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

To commit the file (`moveaces.cmd`) and move the specified user, enter:

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

Remove all ACEs from a access group

Syntax

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

```
remove aces from accgroup GroupName
```

Examples

To remove all ACEs from the HR group, enter In the command file:

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

To commit the file (`moveallaces.cmd`) and remove all ACEs from the specified group, enter:

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

Delete a access group

Syntax

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

```
delete accgroup GroupName;
```

Options

remove_aces=true
If ACEs have not been removed from the group, this option removes all the ACEs as part of the `delete` operation.

Examples

To delete access group `HR` and any ACEs in the group, enter in the command file:

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

To commit the file (`deletegroup.cmd`) and remove the specified user, enter:

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

Create and manage 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` command, 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 Create and manage access control entries on page 69.

Verify administrative authority

Syntax

Prior to making any access control changes, verify the host used for ACL setup has the administrative authority. To verify the host, use the following syntax:

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

Create an access pool

Examples

To create an access pool named `PoolB` in array `12345`, enter In the command file:

```
create accpool PoolB;
```

To commit the file `addnewpool.cmd` and add a new pool, enter:

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

Add devices to access pool

Syntax

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

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

Examples

To assign devices `0A` through `19` to `PoolB`, enter in the command file:

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

To commit the file `addnewdevices.cmd` and devices to the pool, enter:

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

Note

To update the array configuration after Access Control changes are complete, run the `symcfg discover` command. changes

Edit and manage access pools

Once access pools are created, devices can be removed from a pool, or pools can be deleted.

Note

To update the array configuration after Access Control changes are complete, run the `symcfg discover` command.

Remove devices from access pool

Syntax

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

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

Examples

To remove devices `16` through `19` from `PoolB`, enter in the command file:

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

To commit the file `removedevs.cmd` and remove the specified devices from the pool, enter:

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

Delete access pool

Syntax

To delete an access pool from the database, use the following syntax in the command file syntax:

```
delete accpool PoolName;
```

Options

remove_aces=true

Removes any ACEs associated with the pool.

Examples

To delete access pool `PoolB` and any associated ACEs , enter in the command file:

```
delete accgroup PoolB remove_aces=true;
```

To commit the file `deletepool.cmd` and remove the specified user, enter:

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

Create and manage access control entries

Syntax

Access Control Entries (ACEs) grant permissions to groups and pools. Once access groups and pools are created, ACEs are created. A group can have multiple permissions and pools, which requires an ACE for each permission. ACEs determine and grant, to the

access group, the permissions (*AccessType*) for access to a specified pool or all devices not in a pool.

Verify administrative authority

Syntax

Prior to making any access control changes, verify the host used for ACL setup has the administrative authority. To verify the host, use the following syntax:

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

Grant permissions to access group

Syntax

For command the file, use the following syntax:

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

To commit permissions granting, use the following syntax:

```
symacl -sid SymID commit -file FileName
```

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).

Options

AccessType

Specifies permissions to the SYMCLI or SYMAPI features, or functionality granted to selected devices

Available access control permissions

Table 7 Access Control Permissions: AccessType

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
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)

Table 7 Access Control Permissions: AccessType (continued)

Permissions (*AccessType*)	Description	SYMCLI commands affected
		• 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
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

Table 7 Access Control Permissions: AccessType (continued)

Permissions (*AccessType*)	Description	SYMCLI commands affected
		• 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
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›

Table 7 Access Control Permissions: AccessType (continued)

Permissions (*AccessType*)	Description	SYMCLI commands affected
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. Allows for creation of snapvx sessions	symsnap symsnapvx
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

Grant BASE permissions to a group

Examples

Grant BASE permissions to a group before granting permissions to any value-added, non-base feature. To grant BASE permissions to access group `ProdB` for all devices, enter in the command file:

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

To grant BASE permissions to access group `ProdB` for `poolPRODdevs`, enter in the command file:

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

To commit the file `grantbaserights.cmd` and grant BASE permissions, enter:

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

Note

When restricting a host to BASE access to a pool of devices, all devices mapped to the host are 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 are discovered, and the application registration database and the audit logs cannot be accessed since these may contain data relevant to other hosts.

Grant SRDF permissions to a group

Examples

To grant SRDF permissions to access group `ProdB` for pool `PRODdevs`, enter in the command file:

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

To commit the file (`grantsrdfrights.cmd`) and grant SRDF permissions, enter:

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

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).

Grant BCV and SDR permissions to a group

Examples

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

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

To commit the file (`grantbcvsdrrights.cmd)` and grant BCV, and SDR permissions, enter:

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

Grant BASE permissions to a non-registered host

Examples

To make any non-registered (*unknown*) host have BASE permissions to access group UnknwGrp for all devices in the array environment, enter in the command file:

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

To commit the file (`grantbaserights.cmd`) BASE permissions to a non-registered user, enter:

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

Remove permissions from access group

Syntax

In the command file, use the following syntax:

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

To commit removing permissions, use the following syntax:

```
symacl -sid SymID commit -file FileName
```

Obtain access control information

Only ADMIN or ADMINRD permissions allow viewing the access objects (groups, pools, ACLs). Using the list action without administrative positions, display only the access objects associated with the access group for the host executing the `symacl list` command.

List access control information

Syntax

To list information about groups, pools, and ACLs, use the following syntax:

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

Examples

To list the access groups on array `0133`, enter:

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

Show access control information

Syntax

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]
```

Note

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.

Examples
To show the details for access group `ProdB` on array `0133`:

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

Obtain host access ID

Options

-unique
Returns the access ID in a segmented, 24-digit numeric form (`xxxxxxxx-yyyyyyyy-zzzzzzzz`).
For example: `12301558-94200021-00347892`

Examples
To return the access ID for the controlling host, enter:

```
symacl -unique
```

Verify a locked access control session

Examples
To verify an access control session is locked on any array, enter:

```
symacl list -v
```

Reports the session owner and length of time the session has been locked.

Release locked access control sessions

Description
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, the locked session can be aborted.

Syntax
To release the session lock, use the following syntax:

```
symacl release -sid SymmID
```

Note

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

Access control 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 setting up an access control environment for the first time.

Note

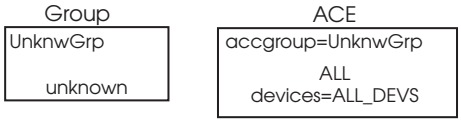
The discover command (`symcfg discover`) must be run after making access control changes.

Default configuration: all permissions to users and hosts

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

When an array is delivered it is configured with a group named *UnknwGrp* created for nonregistered hosts (with no ID), as shown in the figure below.

Figure 2 Example default configuration



A special default access ID named *unknown* is added to the group granting all unknown hosts and users ALL permissions. Next, an ACE is 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.

Manage 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 at array 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.

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 access.

Remove UnknwGrp

Examples

To remove the *UnknwGrp*, enter from the administrative host, enter:

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

Revert UnknwGrp to default

Revert UnknwGrp to default

Examples

To revert access control settings back to the default UnknwGrp, enter from the administrative 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;
```

Limit access to UnknwGrp group

Examples

To limit access to the *UnknwGrp* with limited access (for example BASE permission only to all undefined hosts), enter:

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

Create alternate default ACCID

Description

After removing the default accid *UNKNOWN*, an alternative default accid can be created. A new accgroup is created with permissions that will be the access limitations for all undefined hosts.

Syntax

The default accid provided during setup is called *UNKNOWN*. To create an alternate default accid, use the following syntax:

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

Examples

To create an alternate default accid `Others`, enter:

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

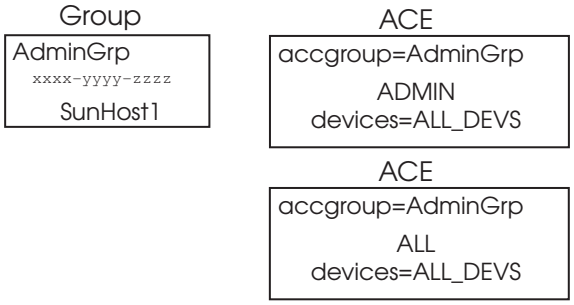
These hosts can discover devices, obtain states and statistics from the array, with BASE permission granted to a default accid `Others`.

Establish an administrator

A newly delivered array allows at least one host assigned with administrative (ADMIN) privileges to the access control database.

In this example, to establish an administrator, an access group named *AdminGrp* is created. Then a UNIX workstation named SunHost1 with an encrypted access ID (of the form *xxxxxxxx-yyyyyyyy-zzzzzzzz*) is added to the AdminGrp group. (Obtain an access ID by running the `symacl -unique` command.) Then two ACEs are created: one granting ADMIN permissions and one granting ALL permissions to group AdminGrp for all devices in the array.

Figure 3 Establish Administrator

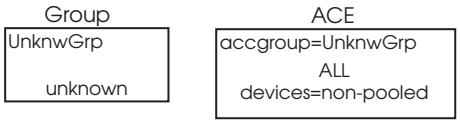


Grant all permissions to the nonpooled devices

Once access controlled access pools have been established, access to all the array devices that are not otherwise registered (*Nonpooled*) in any access pool can be set.

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

Figure 4 Grant all permissions to non-pooled devices



General absolute access control

General absolute access control registers access only to certain devices on an as-needed basis. In this configuration, the *UnknwGrp* group is 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

Once user needs and limitations are determined, it is highly recommended to use the preview and prepare actions on the first major command file before you committing it; particularly if it is an extensive list. The preview and prepare will identify any coding errors or mistakes in the logic.

Setting up access control for TimeFinder devices (5977 and higher)

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

Procedure

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

Setup access control for TimeFinder/Snap devices (Enginuity 5876 or lower)

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

Source and target devices:

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

SAVE devices:

- Define the working access pool to contain the SAVE devices.
- For the group name, grant BASE permissions to access the SAVE devices.
- For the group name, grant CFGSYM permissions to create and delete the SAVE devices.
- For the group name, grant SNAP permissions to manage the SAVE devices.

Setting access control for SRDF devices

To set up an access controlled environment for SRDF operations, set up both the local array and remote array. Because both arrays have their own access controlled database, this requires the following configuration:

Procedure

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).

Setup access control for RecoverPoint Splitter (Enginuity 5876 or lower)

Examples

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

BASECTRL, RPA, and CFGDEV permissions to access group ProdB for pool PRODdevs, enter:

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

Set access control for FAST/Optimizer (Enginuity 5876 or lower)

To set up an access controlled environment for FAST operations, grant permission *OPTMZR* to the host. Obtain the unique access control ID of the array from EMC Customer Support. Add this unique ID 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 is allowed to perform any operations to which the host has been granted rights.

Setup access control for Symaccess

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. This includes the default *UnknwGrp*.

Back up and restore the access control database

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

Create an access control database backup file

Syntax

To create a backup of the current access control database, using the following 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 is compatible for use with the symacl command.

Note

The backup file contains encrypted versions of the unique IDs. Therefore if comparing the values in the backup file to the original file used to create the database, they will be different.

Restore access control database with backup file

Syntax

To restore the previous configuration data to access control database, use the following syntax:

```
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.

Note

The backup file contains encrypted versions of the unique IDs, therefore if comparing 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.

- Device group overview 84
- Device groups 84
- Storage groups 100
- Composite groups 132
- GNS repository 148

Device group overview

Solutions Enabler uses several types of groups to monitor and control VMAX and DMX storage arrays. The type of group depends on your storage, host, and application environment. Grouping devices allows for:

- 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 on page 84).
- 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 on page 100).
- Managing groups of devices spread across multiple local arrays (Composite groups on page 132).
- Ensuring remote data consistency (SRDF consistency groups on page 135).

Device groups

A device group is a user-defined group comprised of devices that belong to a locally attached VMAX or DMX array. Control operations can be performed on the group as a whole, or on the individual device pairs in the group. 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, a device can be added to multiple groups.

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 132.

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.

Group Name Services (GNS) can also be enabled 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 operations, refer to GNS repository on page 148.

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. Assign a device group name when you create it. The name can have up to 31 characters and must be unique for a given configuration database.

When adding 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. This logical name must be used with the `LdevName` or `ld` argument in any SYMCLI command.

Device group types

When creating a device group, define it as 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

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 list types:

- **Standard device list** — Standard device lists provide a mechanism for grouping standard devices in a device group subject to the following restrictions:
  - SRDF and non-SRDF devices cannot be in the same device group *unless* you specify ANY for the type when creating the device group.
  - All SRDF devices in a given device group must belong to the same SRDF group or if concurrent SRDF, belong to two SRDF groups.
- **TimeFinder BCV device list** — TimeFinder BCV device lists provide a mechanism for associating Business Continuance Volume (BCV) regular devices and RDF1 BCV devices with a device group. BCV control operations can be performed 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. Virtual devices (VDEV) paired with standard and BCV devices can be associated 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 Device restrictions on page 86) 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** — One or more gatekeeper devices can be associated 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. A standard device can be added 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.

Device restrictions

Restrictions on the devices you can add to a device group are based on the device type.

RDF1 and RDF2 type device restrictions

The following restrictions apply when adding an SRDF device to a device group:

- All devices in the device group must be SRDF devices.
- All devices in the device group must be either all source (RDF1 type) or all target (RDF2 type) devices.
- All devices in the device group 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 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 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. The following 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 Export and import device lists on page 93.

Create a device group

Create a device group by defining a named empty group of a specific type and then adding devices to the group. A newly-created device group is defined by the devices added to it. Each type of device list has its own set of restrictions.

Create an empty device group

Syntax

To create a device group, use the following syntax:

```
symdg -type GroupType create GroupName
```

Options

- type

Specifies the group type. Valid group types are: `REGULAR, RDF1, RDF2, R21` or `ANY`. The default type is `Regular`.

name
Specifies the group name. The name can have up to 31 characters and must be unique for a given configuration database.

Example

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.

Add devices to a device group

Description

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`). A logical device name can also be assigned to a device. A valid logical device name cannot exceed 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.

Note

If the device group already contains one or more storage groups, the command to add devices fails as this action violates the restrictions for device groups containing storage groups.

Syntax

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
```

Options

SymDevName
Specifies the device name when adding virtual devices or target devices.

Interval
The time to wait between attempts to acquire an exclusive lock on the host database.

Count
The number of attempts.

Examples

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
```

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
```

Add virtual devices to a device group

Syntax

The following is the syntax for adding a virtual device:

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

Options

-vdev
Specifies that the added device is a virtual device.

-rdf
When adding virtual devices from a remote array (RVDEV), targets the operation to the specified virtual device over SRDF links on the remote array.

-rdfg
With concurrent SRDF, specifies the SRDF group number (`-rdfg GrpNum`), with the `-rdf` option.

-remote_rdfg, -hop2
Targets the operation to the specified virtual device over SRDF links 2 hops away.

Example

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
```

For example, to add virtual device 00005 to a device group named **prod1** from a remote array 2 hops away, enter:

```
symdg -g prod1 add dev 00005 vdev1 -vdev -hop2 -remote_rdfg 10
```

Add devices to the target list

Description

For TimeFinder/Clone operations, devices can be added to the target device list (TGT) of a device group or the remote target device list (RTGT). STD, SRDF, BCV, and VDEV devices can be added to the TGT, RTGT, and Hop2 TGT target lists.

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 86.

Options

-vdev
Specifies that the added device is a virtual device.

-tgt
Specifies that added devices are from a local array.

-rdf
When adding virtual devices from a remote array (RVDEV), targets the operation to the specified virtual device over SRDF links on the remote array.

-rdfg

With concurrent SRDF, specifies the SRDF group number (`-rdfg GrpNum`), with the `-rdf` option.

-remote_rdfg, -hop2
Targets the operation to the specified virtual device over SRDF links 2 hops away.

Examples
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
```

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
```

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 TGT*xxx*.
- VDEVs added to the remote target list are named RTGT*xxx*.
- VDEVs added to the Hop-2 target list are named 2TGT*xxx*.

Add ungrouped devices to a device group

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

Syntax
By default, all standard devices (or local virtual devices) are added to a device group, unless options are used to specify certain types of devices. To add ungrouped devices to a device group, use the following syntax:

```
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 . . .]
```

Options

-i
The time to wait between attempts to acquire an exclusive lock on the host database.

-c
The number of attempts.

-sid
All ungrouped devices from a specific array ID.

-vdev -rdf -rdfg GrpNum
All virtual devices from a local or remote array.

-tgt -rdf -rdfg RemoteGrpNum
All devices are added to the target list of the device group for TF/Clone operations on a remote array.

-vdev -hop2 -remote_rdfg RemoteGrpNum
All virtual devices from a remote array two hops away.

-tgt -rdf -hop2 -remote_rdfg RemoteGrpNum
All devices are added to the target list of the device group for TimeFinder/Clone operations on an array two hops away.

-SA ‹#|ALL›
All devices visible to one or all front-end directors.

-P #
All devices visible to one or all front-end port number.

-N #
The number of devices to add to the device group.

-devs
Any combination of device ranges and single devices, such as 225:22a, 120,a5,7a0:7af

-CAP #
Any combination of device ranges and single devices that are of a specific capacity.

-sel_rdfg
Only SRDF devices belonging to that group number are added.

Note

When using concurrent SRDF, when 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`.

Restrictions
The following are not allowed when using `addall` command:

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

Set controls on Celerra devices

Syntax
To set controls on Celerra FBA devices with the `symdg -celerra` command, use the following syntax:

```
symdg -g <DgName> [-noprompt] [-i <Interval>] [-c <Count>]
[-bcv | -vdev | -tgt] [-rp] [-star] [-celerra]
                rw_enable [LdevName [LdevName...]] [-p <#>] [-SA <#|
ALL>]
                write_disable [LdevName [LdevName...]] [-p <#>][-SA
<#|ALL>]

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

```
                not_ready [LdevName [LdevName...]]
```

List devices in a device group

Syntax

To list devices in a device group, use the following syntax:

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

List all device groups

Description

Use the `symdg list` command to list all device groups defined in the configuration database, including group names, group type, array ID, the number of standard, BCV, virtual devices, and TF/Clone target devices (TGTs). The output also indicates if the device group is valid and whether it is contained by a composite group.

Options

-novalid

Eliminates the validation of groups during the execution of the `list` command. The `V` column under the `Flags` field and the `V` description in the legend do not display in the output.

Example

```
symdg list
```

Note

Device groups with names longer than 17 characters display with their first 17 characters followed by an asterisk (*). Display composite group information on page 144 explains how to set the environmental variable to display longer group names.

Sample output

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

Show device group details

Description

Use the `symdg show` command to display information about a specific device group.

Example

```
symdg show dgincg
```

Sample output

```
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 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
....
```

Export and import device lists

The list of devices from an existing group can be saved to a file on the host system, and this file can later be imported to create a device group. Device list files are used to recreate a device group that was deleted, or for importing the group into another system.

Export a device list

Syntax

To remove all devices from a group but retain or export the list of devices in the group to a file on your host system, use 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>]
```

Options

-rdf
Exports an SRDF group. Uses the remote array ID and device names and changes the SRDF group type from R1 to R2 or R2 to R1.

-rdfg GrpNum
Specifies an SRDF group number.

-delete
Exports the device group membership to a file and deletes the existing device group in the same operation. To reinstate the same group again, import this list to the same or different device group. Refer to Import a device list on page 94.

Example

To create a text file that contains the details of all members of the existing device groups, use the `symdg -exportall` operation. To later recreate the device groups from this file, use the `symdg importall` command.

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

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

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 Delete a device group on page 99.

Note

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

Import a device list

Description

Typically imported files were previously exported (refer to Export a device list on page 93).

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

To recreate all device groups, use 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.

Syntax

To add multiple devices to a new or existing device group by importing an existing file, use the following syntax:

```
symdg -sid SymmID [-i Interval] [-c Count][-v]
import DgName [-f FileName]
importall [-f FileName]
```

Example

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

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

Rename device groups

Description

Use the `symdg rename` command 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.

Options

-v

For a device group that is a member of one or more composite groups, use the verbose option to view associated composite groups.

Examples

To rename the device group `prod` to `prod_B`, enter:

```
symdg rename prod prod_b
```

To rename a device group that is a member of one or more composite groups, enter:

```
symdg rename prod prod_b -v
```

Sample output

Using verbose option:

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

Rename logical device names

Description

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

Note

This command fails if attempting to rename a logical device name of a device in a device group containing storage groups.

Example

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

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

Move a device between device groups

Description

Use the `symdg` command to move one device from one device group to another. The source and destination device groups must be compatible types.

Note

This command cannot be used to move devices from a device group containing storage groups. Use the `symsg move` or `symsg moveall` commands to move devices in a storage group.

Syntax

To move an individual device, use the following syntax:

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

Options

-i

Specifies the predetermined time (interval) to wait between attempts to acquire an exclusive lock on the array host database and, for SRDF control operations, on the local and/or remote arrays.

-c

Specifies the number of attempts (count).

-rename

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, use the –`rename` option to avoid encountering an error. When this option is used, SYMCLI renames the moved device to the next available logical device name as defined in the SYMCLI_LDEV_NAMING environment variable.

Example

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

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

Move all devices between device groups

Description

Use the `symdg moveall` command to move all standard devices from one device group to another. The source and destination device groups must have compatible device types.

Syntax

To move all devices, use the following syntax:

```
symdg -g DgName[-i Interval] [-c Count][-v] [-offline]
moveall DestDgName [-force] [-rename]
[-vdev | -tgt [-hop2] | -rvdev | -rtgt]
```

Options

-i

Specifies the predetermined time (interval) to wait between attempts to acquire an exclusive lock on the array host database and, for SRDF control operations, on the local and/or remote arrays.

-c

Specifies the number of attempts (count).

-rename

If there is a device within the destination device group with the same logical device name as the device being moved, use the –`rename` option to avoid an error. When this option is used, SYMCLI renames the moved device to the next available logical device name as defined in the SYMCLI_LDEV_NAMING environment variable.

-vdev

Moves only the virtual devices to the destination device group.

-hop2

Indicates that the specified device is two hops away.

-tgt

Moves only the TGT devices to the destination device group.

-rdev

Moves only remote virtual devices to the destination device group.

-rtgt

Moves only RTGT devices to the desitination device group.

Example

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

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

Copy devices between device groups

Description

Devices from an existing device group can be copied into another existing device group of compatible type. Use the `copy` action to copy one standard device from the specified source device group to the destination device group. The source and destination device groups must have compatible types.

Use the `copyall` action to copy all standard devices 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), the option file setting `SYMAPI_ALLOW_DEV_IN_MULT_GRPS = ENABLE` is required.

Syntax

To copy devices from one device group to another device group, use the following syntax :

```
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]
```

Copy devices from device group to storage group

Description

Devices from a 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*).

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

Note

The `symdg dg2sg` command is blocked for device groups containing storage groups.

Syntax

To add devices from a device group to a storage group, use the following syntax:

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

Example

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

```
symdg dg2sg prod prod_2
```

Remove a device from a device group

Description

Use the `symdg remove` command to remove a device from a device group. By default, the `remove` argument affects standard devices only. However, the `-vdev` option can be specified to remove a virtual device.

Note

If you remove the only member of a device group, the device group is not automatically deleted. Use the `symdg delete` command to explicitly delete the device group, as described in the Delete a device group on page 99. If the device is contained in a storage group, the operation fails. Use the `symsg remove dev` command to remove a device in a storage group.

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.

Remove all devices from a device group

Description

Use the `symdg rmall` command to remove all devices from a device group. By default, the `rmall` argument affects standard devices only. However, the `-vdev` option can be specified to remove just the virtual devices.

Note

Removing all members of a device group, does not automatically delete the device group. Use the `symdg delete` command to explicitly delete the device group, as described in Delete a device group on page 99. If the devices are contained in a storage group, the operation fails. Use the `symsg rmall` command to remove devices in a storage group.

Syntax

To remove all devices from an existing device group, use 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

-SA ‹#|ALL›
Specifies devices mapped to a specific front-end (SCSI or Fibre) director number.

-P #
Specifies devices mapped to a specific (SCSI or Fibre) director port number.

-CAP #
Specifies devices of a specified capacity.

-devs
Specifies any combination of device ranges and single devices to remove.

-vdev | -tgt -rdf [-rdfg GrpNum]
Specifies local and remote virtual and target devices.

Delete a device group

Syntax

To delete a device group, use the following syntax :

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

Examples

To delete a device group named `prod`, enter:

```
symdg delete prod
```

When deleting device groups that are associated with a composite group, enter the verbose (`-v`) option to view the composite groups:

Note

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

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

Perform control operations on device group devices

Description

Use the `symdg` command 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
- Set or unset Geometry Compatible Mode (GCM)

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

- Compress or uncompress a thin device
- Set the Optimized Read Miss mode (ORM)

Note

For arrays running Enginuity 5876 Q22013SR, 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, use the appropriate qualifier (such as, -bcv, -rbcv, etc.).

- Bind, unbind, rebind, start or stop allocations, free, and reclaim thin devices

Note

Free and reclaim actions are blocked for any device which is the source or target of a Clone, Emulation, or Snap session.

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 is performed on devices in a specific list, the appropriate qualifier needs to be used (such as, `-bcv`, `-rbcv`, etc.).

Note

Virtual Pool (VP) compression (Enginuity 5876 only) on page 353 provides information on compressing and uncompressing thin devices. For more information on actions for thin devices (bind, unbind, rebind, allocate, etc.) refer to Perform actions on thin devices (Enginuity 5773 through 5876) on page 341.

Syntax

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]
```

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. Use the SYMCLI **symsg** command to create and manage these storage groups.

Storage group 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:

- Storage groups are supported only on VMAX arrays running Enginuity version 5874 to 5876 or HYPERMAX OS 5977.
- GNS does not support storage groups. Storage groups are saved in a special area on the array.
- For arrays running Enginuity 5874 to 5876, the maximum for a single array is 8192 storage groups. For arrays running HYPERMAX OS 5977, the maximum for a single array is 16K storage groups.
- Each storage group can contain a maximum of 4096 devices.
- 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 Add cascaded storage groups on page 111 and Restrictions for storage groups with defined Host I/O limits (5876 or 5977) on page 114.

Create storage groups

Syntax

To create and empty storage group, 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>]
```

Options

bw_max
Specifies 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
Specifies the I/Os per second of the devices over a set of director ports. The valid range for IOPs is from 100 IO/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

Host I/O Limits for storage groups (5876 or 5977) on page 114 describes how to use the bw_max, iops_max, and dynamic options to set Host I/O limits.

-slo

Specifies the Service Level Objective for the storage group as follows in the order of the highest to lowest performance expectation:

- Diamond – emulates EFD performance.
- Platinum – emulates between EFD and 15K drive performance.
- Gold – emulates 15K drive performance
- Silver – emulates 10K drive performance
- Bronze – emulates 7.2K drive performance
- Optimized – Balances performance across the whole SRP, based on I/O load, type of I/Os, data pool utilization, and available capacities in the pools. It places the most active data on higher performing storage and least active data on the most cost-effective storage. Optimized SLO does not use a workload type.

Note

If no SLO is specified then default SLO *Optimized* is set for the storage group.

-srp

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

-wl

Specifies the workload type as follows:

- OLTP – Online Transaction Processing
- DSS – Decision Support System

Note

If Workload is not specified then a value of *none* is assigned.

The Service Level Objective and Storage Resource Pool name parameters behave as follows:

- If the *SLO Name* and *SRP Name* are not specified, the storage group is created with no SLO or SRP and it is not FAST-managed.
- If both *SLO Name* and *SRP Name*, are specified, the storage group is FAST-managed.
- If only the *SLO Name*, is specified, a default Storage Resource Pool for the emulation type of the devices in the storage group is used and the storage group is FAST-managed.
- If only the *SRP Name* is specified, an Optimized Service Level Objective is used with the storage group and it is 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.

Create devices and add to storage groups

Note

For HYPERMAX OS 5977, 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.

Syntax

To add a device to a storage group, use the following syntax (5876 or lower):

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

To add a device to a storage group, use the following syntax (5977 or higher):

```
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>]]
     ...
```

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.

Add existing devices to storage groups

Syntax

To add a device, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
          add dev <SymDevName>
```

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

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

Options

-tgt

Specifies adding only target devices listed in a text file. Device text files are either a one or two column format; source devices listed in the first column and target devices listed in second column.

Examples

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

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

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

```
symsg -sid 123 -sg prod addall pd
```

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

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

To a dd a range or list of devices to storage group `prod` on array `123`:

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

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

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

Note

Any device that belongs to storage group that is part of a masking view cannot be added to another storage group.

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.

Add child storage groups to existing storage groups

Syntax

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

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
          add sg <SgName1>[,<SgName2>,<SgName3>,...,<SgNamen>]]
```

Restrictions

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

Modify storage group properties

Syntax

To modify the storage group properties, 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]
```

Options

bw_max
: Specifies 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
: Specifies 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

Host I/O Limits for storage groups (5876 or 5977) on page 114 describes how to use the bw_max, iops_max, and dynamic options to set Host I/O limits.

-slo

Specifies the Service Level Objective for the storage group as follows in the order of the highest to lowest performance expectation:

- Diamond – emulates EFD performance.
- Platinum – emulates between EFD and 15K drive performance.
- Gold – emulates 15K drive performance
- Silver – emulates 10K drive performance
- Bronze – emulates 7.2K drive performance
- Optimized – Balances performance across the whole SRP, based on I/O load, type of I/Os, data pool utilization, and available capacities in the pools. It places the most active data on higher performing storage and least active data on the most cost-effective storage. Optimized SLO does not use a workload type.

Note

If no SLO is specified then default SLO *Optimized* is set for the storage group.

-noslo

Specifies a Service Level Objective to be removed from a storage group.

-wl

Specifies the workload type as follows:

- OLTP – Online Transaction Processing
- DSS – Decision Support System

Note

If no workload is specified then default workload *none* is set for the 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.

Set SLO for a storage group

Description

When setting a SLO for a storage , a workload can also be set. If no workload is specified, then the workload *none* is assigned. If the storage group does not have a SRP set, the system default SRP for the emulation type of the devices in the storage group (FBA or CKD) is used and the group becomes FAST-managed.

Syntax

To set a SLO for a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-slo <SLO Name> [-wl <Workload Name>]
```

Options

-slo

Specifies the SLO for the storage group as follows in the order of the highest to lowest performance expectation:

- Diamond – emulates EFD performance.
- Platinum – emulates between EFD and 15K drive performance.
- Gold – emulates 15K drive performance
- Silver – emulates 10K drive performance
- Bronze – emulates 7.2K drive performance
- Optimized – Balances performance across the whole SRP, based on I/O load, type of I/Os, data pool utilization, and available capacities in the pools. It places the most active data on higher performing storage and least active data on the most cost-effective storage. Optimized SLO does not use a workload type.

Note

If no SLO is specified then default SLO *Optimized* is set for the storage group.

-wl

Specifies the workload type as follows:

- OLTP – Online Transaction Processing
- DSS – Decision Support System

Note

If no workload is specified then default workload *none* is set for the storage group.

Remove SLO from a storage group

Description

When removing the SLO from a storage group, any workload that was assigned for the SLO is removed. If the storage group has a SRP, the storage group will be assigned an *Optimized* SLO. Otherwise, there is no SLO or SRP for the storage group and the group will no longer be FAST-managed.

Syntax

To remove the SLO from a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> -noslo
```

Set SRP for a storage group

Description

When setting a SRP for a storage group. if the storage group does not have an assigned SLO, an *Optimized* SLO is assigned and the group becomes FAST managed.

Syntax
To set the SRP for a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> -srp <SRP Name>
```

Remove SRP from a storage group

When removing an SRP from a storage group, if the storage group has an assigned SLO, the system default SRP for the emulation type of the devices in the storage group is used. If there is no assigned SLO, the storage group is no longer FAST-managed.

Syntax
To remove the SRP from a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> -nosrp
```

Remove SLO and SRP from a storage group

Syntax
To remove the SLO and SRP from a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> -noslo -nosrp
```

Note

When the SLO and SRP are removed from a storage group, it is no lonter FAST-managed.

Change workload on a storage group

Description
To change the workload without changing the assigned SLO, specify both the `-slo` option with the current SLO name and the `-wl` option with the new workload name.

Syntax
To change the workload type for a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> -slo <SLO Name>
-wl <Workload Name>
```

Remove workload on a storage group

Description
To remove the workload on a 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.

Syntax
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]
```

Restrictions for storage group modification

- 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.

Standalone storage groups and cascaded groups

Note

This topic applies only to arrays running Enginuity 5876 or HYPERMAX OS 5977 or higher.

Solutions Enabler 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.

Parent and child storage group restrictions

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` operations affect all of the devices in all of the child storage groups contained by that parent:

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

Add cascaded storage groups

Note

This topic applies only to arrays running Enginuity 5876 or HYPERMAX OS 5977 or higher.

Description

Use the **symsg add sg** command to add child storage groups individually to a parent storage group.

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.

Syntax

To add child storage groups to a parent storage group, use the following syntax:

```
symsg -sg SgName -sid SymmID [-i Interval]
          [-c Count] [-v] [-celerra] [-rp] [-ckd]
        add sg SgName1[,SgName2,SgName3,"¦,SgNamen
```

Restrictions

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

- 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 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, the operation will fail.

Convert standalone storage group to cascaded group

Note

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

Use the `symsg convert -cascaded` CLI command to non-disruptively convert from a standalone storage group to 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 it must be specified, 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.

Syntax

To convert a storage group to a cascaded group, use the following syntax:

```
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 user-supplied storage group must exist.
- The supplied child storage group must not already exist.
- The supplied storage group must be standalone.
- If the supplied storage group has a Host IO limit defined, the **host_IO** option must be specified .

Convert cascaded group to standalone storage group

Note

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

Use the `symsg convert -standalone` CLI command to non-disruptively convert from a cascaded storage group consisting of a parent storage group and a single child storage group to 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, the **host_IO** option must be supplied. A storage group has Host I/O Limits set if a limit to either `bw_max` or `iops_max` was configured on the group.

Syntax

To convert a cascaded group to a standalone storage group, use the following syntax:

```
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.

Remove devices from a storage group

Description

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

Options

-tgt

Specifies adding only target devices listed in a text file. Device text files are either a one or two column format; source devices listed in the first column and target devices listed in second column.

Syntax

To remove a single device from a storage group, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
         remove dev <SymDevName> [-force]
```

To remove multiple devices, devices in a range, or a file, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
          [-SA <# | ALL>] [-p <#>] [-N <#>]
          [-cap <#> [-captype <mb> | <cyl>]]
          [-devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
            [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>>...]> |
           -file <DeviceFileName> [-tgt] ]
           rmall [-force]
```

Example

To remove all devices that are listed in text file `storgrp_a.txt` from storage group `prod` on array `123`, enter:

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

Note

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

Remove child storage groups

Description

To remove child storage groups individually from a parent storage group, use the **symsg remove sg** command.

Syntax

Use the following syntax to remove 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]
```

Restrictions

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.

Host I/O Limits for storage groups (5876 or 5977)

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 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.

Host I/O Limits can be added, removed, or modified for a storage group. 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

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 (5876 or 5977)

Note

The Host I/O Limits feature requires Enginuity 5876 Q42012SR or HYPERMAX OS 5977.

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.

Set the Host I/O Limit (5876 or 5977)

Description

Note

The Host I/O Limits feature requires Enginuity 5876 Q42012SR or HYPERMAX OS 5977.

The **symsg create** command provides two options for setting the storage group Host I/O Limit (**-bw_max** *MBperSec* and **-iops_max** *IOperSec*). To configure Host I/O Limits on a set of devices, the front-end 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.

- 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.

Example
To create an empty storage group named **prod** with a front-end bandwidth limit of 40,000 MB per second on array ID 123, enter:

```
symsg -sid 123 create prod -bw_max 40000
```

Change the Host I/O Limits (5876 or 5977)

Description
The Host I/O Limits feature requires Enginuity 5876 Q42012SR or HYPERMAX OS 5977. The same range limits described in Set the Host I/O Limit (5876 or 5977) on page 115 apply, but if `NOLIMIT` is specified, then the maximum bandwidth or IOs/Sec is set to unlimited. List storage groups on page 118 provides more information on listing storage group information including Host I/O Limits feature and demand reports.

Note

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 with FCoE ports.

Syntax
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]
```

Example
To set an IOPs limit of `50,000` I/Os per second for a storage group named `prod` on array ID `123`, enter:

```
symsg -sg prod -sid 123 -sg prod set -iops_max 50000
```

Set dynamic distribution for FE Host I/O limits (5876 or 5977)

Description

Note

The Host I/O Limits feature requires Enginuity 5876 Q42012SR or HYPERMAX OS 5977.

You can optionally configure a dynamic distribution of the FE Host I/O limits by using the **-dynamic** option with the **symsg set** and **symsg create** commands. If the option field is not specified, a default of NEVER is used.

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.

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.

Syntax

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>]>
```

Options

-dynamic

- ALWAYS: indicates that the Host IO Limits for the storage group should always be dynamically redistributed
- NEVER: indicates that the Host IO Limits for the storage group should never be dynamically redistributed
- ONFAILURE: indicates that the Host IO Limits for the storage group should dynamically redistributed only upon a failure of a Front-End Port

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.

Copy devices from a storage group to a device group

Description

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*).

Syntax

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]
```

This command 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.

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.

Example

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
```

List storage groups

Description

The `symsg list` command returns a list of all storage group names and the following details:

- Number of sevices
- Number of gatekeepers
- Number of child storage groups — For parent storage groups, displays both the number of child storage groups it contains and the cumulative total of all devices contained in the child storage groups.
- FAST association
- Masking view status
- Cascade status
- Host I/O Limits status

Note

Storage groups with names longer than 21 characters display with their first 21 characters followed by an asterisk (*). Display full group names on page 32 explains how to set the environmental variable to display full group names.

Syntax

To retrieve a list of all storage groups for a specified array, use the following syntax:

```
symsg list -sid SymmID
```

Example

To display all storage groups for array `230`, enter:

```
symsg -sid 230 list
```

Note

For the (M)asking View flag, if a parent storage group contains a masking view, all its child storage groups display that they are contained within the same masking view.

Sample output

```
 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
```

Report Host I/O Limit demands (5876 or 5977)

Description

Note

Enginuity 5876Q42012SR or HYPERMAX OS 5977 is required for this feature.

Use 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.

- On arrays running Enginuity 5876, all front-end emulations support a fixed number of ports (up to two). 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.

Syntax

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
```

Examples

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.

Show storage group details

Examples

To show information about storage group `SG_Corp1` on array `601`, enter:

```
symsg show -sid 601 SG_Corp1
```

Sample output

This output shows a storage group without a defined Host I/O Limit.

```
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
      . . .
```

Show cascaded storage group details (Enginuity 5876 or HYPERMAX OS 5977)

Examples

To show information for parent storage group `SG_Eng1` for array `601`, enter:

```
symsg show SG_Eng1 -sid 601
```

To show information for child storage group `Eng1_Data` for array `601`, enter:

```
symsg show Eng1_Data -sid 601
```

Sample output

For parent storage group:

Note

The `Number of Composite Groups` and `Number of Groups` fields display as `0` for a child storage group whose parent belongs to a composite group or device group. Only a storage group that is directly contained by the composite group or device group displays a value.

```
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
      . . .
```

For child storage group:

Note

This example shows a child storage group 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.

```
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
  . . .
```

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
```

Export and import device lists

You can export 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:

```
SymmIDSymDevNameSymDevName. . .
```

Group files for the **importall** and **exportall** commands contain device parameters in the following format:

```
SgNameSymmIDSymDevNameSymDevName. . .
```

Repeat this format 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 command is executed.

Export a device list

Description

Use the **symsg export** or **exportall** to export a list of devices from one or all storage groups to a text file to the host system.

For arrays running HYPERMAX OS 5977, **symsg export** and **exportall** support exporting the Service Level Objective, the workload, and the Storage Resource Pool, 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.

Note

If the storage group configuration, as specified in the export file, has a Storage Resource Pool set and if the Storage Resource Pool has been renamed between the export and import operations, the storage group will not be imported.

Syntax

To export a storage group device list, use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count][-v]
 export <SgName> [-file <FileName>] [-offline
 exportall [-file <FileName>] [-offline]
```

Examples

To export the storage group membership from group `prod2` to device file named `prod2list.txt`, enter:

```
symsg -sid 123 export prod2 -f prod2list.txt
```

Sample output

The output of the text file with the exported data displays as follows:

Note

Identifier (`S`), denotes a child storage group name.

```
000194900341
S sgchild1
S sgchild2
```

The output of a text file generated from `symsg exportall` displays as follows:

```
<sgparent>
000194900341
S sgchild1
S sgchild2
<sgchild1>
000194900341
00201
00206
<sgchild2>
000194900341
00404
00405
00406
```

Import a device list

Description

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. 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 are attempting to create already exist, Solutions Enabler displays a message indicating it exists, 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.

Note

For arrays running Enginuity 5773 through 5876, the Service Level Objective name, workload, Storage Resource Pool name, and RDF coordination will not be copied.

For arrays running HYPERMAX OS 5977, **symsg import** and **importall** support importing devices from the FAST-managed storage group in the import file to a target storage group as follows:

- If the imported storage group exists in the array, the target storage group will have its Service Level Objective, workload, and Storage Resource Pool 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.
- The `import` fails if the storage group is configured with a SLO other than "Optimized" and the SRP set for the storage group is configured for FTS external provisioning. With an SLO other than "Optimized" and an SRP that is configured for FTS external provisioning `importall` skips the import of these storage groups.

Syntax

Use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count][-v]
import SgName [-f FileName]
importall [-f FileName
```

Example

To create a storage group named **prod2** from the device file **prod2list.txt**, enter:

```
symsg -sid 123 import prod2 -f prod2list.txt
```

Rename a storage group

Syntax

To rename a storage group, use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count]
    rename OldSgName NewSgName-v
```

Note

If the storage group is contained by a composite group or device group, all local device groups containing the storage group updatewith the new name of the storage group. The update is then propagated to the GNS or RDF daemons and the SYMAPI database.

Example

To rename a storage group named `prod` on array `123` to `prod_B`, enter:

```
symsg rename -sid 123 prod prod_B
```

Move devices between storage groups

Use the **symsg move dev** or **symdev moveall** command to move one or all devices from one existing storage group to another existing storage group.

Conditions

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

  If 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.

Move a single device between storage groups

Description

Use the **symsg move dev** command to move one device, specifying the device name, from one storage group to another 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.

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.

The **-force** flag is not required for non-disruptive move operations.

Syntax

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]
```

Example

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.

Move all devices between storage groups

Description

Use the `symsg moveall` command to move multiple devices from a storage group in a list, using a combination of device ranges and single devices, or from a text file. The moved devices are deleted from the current storage group (`SgName`) and added to the destination storage group (`DestSgName`).

When moving 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.

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.

Syntax

To move multiple devices, devices in a range, or listed in a file, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
          [-SA <# | ALL>] [-p <#>] [-N <#>]
          [-cap <#> [-captype <mb> | <cyl>]]
          [-devs <<<SymDevStart:SymDevEnd> | <SymDevName>
           [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]> |
           -file <DeviceFileName> [-tgt] ]
           moveall DestSgName
```

Examples

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 test
```

Restrictions

- For HYPERMAX OS 5977, selected or all the devices in a source storage group can be moved to a standalone or child destination storage group that is FAST-managed. However, the operation fails if after the move the device is in more than one FAST-managed storage group, or if moving encapsulated devices to the storage group.
- For Enginuity versions 5773 through 5876, moving all devices out of a storage group that is part of a masking view or associated with a FAST policy is not allowed.

Copy a device between storage groups

Move all devices between storage groups

Description

Use the `symsg copy dev` command to copy one device, specifying the array device name, from one storage group to another storage group. The copied device remains in the current storage group (`SgName`) and is added to the destination storage group (`DestSgName`). 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.

Syntax

To copy an individual device, use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]
          copy dev <SymdevName> <DestSgName>
```

Example

To copy device `30` on array `123` in storage group `prod` to storage group `test`, enter:

```
symsg -sid 123 -sg prod copy dev 30 test
```

Restrictions

- For HYPERMAX OS 5977, a device in a source storage group can be copied to a standalone or child destination storage group that is FAST-managed. However, the operation fails if the device is already in another storage group that is FAST-managed or if copying encapsulated devices to the storage group.
- For Enginuity versions 5773 through 5876, copying a device from a storage group that is associated with a FAST policy into another storage group or associated with a FAST policy is not allowed.

Copy all devices between storage groups

Description

Use the `symsg copyall` command to copy multiple devices, from one storage group to another storage group. Multiple devices are 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 is added to the destination storage group (`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.

Syntax

To copy multiple devices from a storage group, use the following syntax:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-v]
        [-celerra] [-rp] [-ckd][-SA <# | ALL>] [-p <#>] [-N <#>]
  [-cap <#> [-captype <mb> | <cyl>]]
        [-devs <SymDevStart:SymDevEnd | SymDevName >
        [<,SymDevStart:SymDevEnd | SymDevName...]> |
        -file DeviceFileName [-tgt]]
        copyall DestSgName
```

Examples

To copy all devices from a storage group named `prod` to a storage group named `test` on array `123`, enter:

```
symsg -sid 123 -sg prod copyall 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 test
```

To copy multiple devices from device file `test_2.txt` to storage group `test` and destination storage group `test_2` on array `123`, enter:

```
symsg -sid 123 -sg test copyall -file test_2.txt test_2
```

Restrictions

- For HYPERMAX OS 5977, all the devices in a source storage group can be copied to a standalone or child destination storage group that is FAST-managed. However, the command fails if the device is already in another storage group that is FAST-managed or if copying encapsulated devices.
- For Enginuity versions 5773 through 5876, copying devices from a storage group that is associated with a FAST policy into another storage group that is also associated with a FAST policy is not allowed.

Copy devices from a storage group to a device group

Description

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*).

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

Syntax

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]
```

Examples

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
```

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
```

Note

An error will be returned if the -tgt option is specified and the storage group contains both standard and BCV devices.

Delete a storage group

Syntax

Use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count] delete SgName [-force] -v
```

Note

Use the **-force** option to force the deletion of a storage group that is contained by a device group or composite group. On the host performing the delete, all device groups or composite groups containing the deleted storage group are updated. The update is then propagated to the GNS daemon and the database. Deleting a storage group contained by a composite group causes the RDF daemon to stop monitor the composite group.

Example

To delete an empty storage group named `prod` on array `123`, enter:

```
symsg delete -sid 123 prod
```

Restrictions

- Deleting a storage groupthat is part of a masking view or associated with a FAST policy is not allowed.
- 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.

Perform control operations on storage group devices

Description

Use the **symsg -sg** command 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

- Sets or unsets Geometry Compatible Mode (GCM)

Note

For arrays running HYPERMAX 5977, the Optimized Read Miss feature is not supported.

- Set the Optimized Read Miss mode

Virtual Pool (VP) compression (Enginuity 5876 only) on page 353 provides information on compressing and uncompressing thin devices.

For more information on actions for thin devices (bind, unbind, rebind, allocate, etc.) refer to Perform actions on thin devices (Enginuity 5773 through 5876) on page 341.

For arrays running Enginuity 5876 Q22013SR, the **symsg** command allows you to set the Optimized Read Miss mode to Off or System Managed for all devices in a storage group.

Syntax

Use the following syntax:

```
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]
```

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.

Composite group 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 within composite groups

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 85 and Create a device group on page 87.

Logical names of devices within 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:

  Table 8 Logical name formats

2BCV	BRBCV	RTGT
2TGT	DEV	RVDEV
2VDEV	RBCV	TGT
BCV	RRBCV	VDEV

- *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

Use one of the following types to create a composite group:

- 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`) 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 (ELN) which can be reached at http://elabnavigator.EMC.com.

Create a composite group

To create a composite group:

1. Define a named empty group of a specific type explained in Composite group types on page 134.
2. Add devices OR device groups to the composite group.

Create an empty composite group

Description

Use the `symcg` command to create a composite group. When you create a composite group, you assign it a name and a group type.

Syntax

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]
```

Options

-i
Specifies the interval to wait between attempts to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

-c
Specifies the number of attempts to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

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

Create a composite group from devices in a storage group

Description

Starting with Solutions Enabler V8.0.1, you can 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.

Syntax

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]
```

Options

sg2cg
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.

Create composite groups with SRDF consistency

Description

Specify the `-rdf_consistency` option parameter with `symcg create` to register a composite group of type RDF1, RDF2, or RDF21 with the SRDF daemon.

Note

Before the SRDF daemon can begin monitoring and managing a composite group created with SRDF consistency, enable it using the `symcg enable` command. Note that 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).

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.

Examples

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
```

Set controls on Celerra devices

Syntax

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]
```

Export a composite group to file

Syntax

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>]
```

Options

-i

Specifies an interval to wait between attempts to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

-c

Specifies the number of attempts to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

-rdf

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.

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.

Delete a composite group

Syntax

Use the following syntax to delete an existing composite group:

```
symcg [-i Interval] [-c Count][-v]
   delete CgName [-force] [-symforce]
```

Options

-force

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.

Import a composite group

Syntax

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]
```

Options

-rdf_consistency

When the `-rdf_consistency` parameter is specified, the composite group is registered with the SRDF daemon.

-rename

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.

Add standard devices to a composite group

Syntax

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]
```

Note

If the composite group already contains one or more storage groups, the command to add devices will fail as this action violates the restrictions for device groups containing storage groups.

Options

-i
Specifies an interval to wait between attempts to acquire an exclusive lock on the host database on the local and/or remote arrays.

-c
Specifies the number of attempts to acquire an exclusive lock on the host database on the local and/or remote arrays.

-sid
Specifies the array ID when adding devices by specifying a device name.

-rdf
When specified with the `-vdev` option, adds the devices to the remote virtual device list. When specified with the `-tgt` option, adds the target devices to the remote target device list.

-hop2
Same as `-rdf`, but when the device is two hops away.

-rdfg
Specifies the SRDF group number.
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.

-vdev
When the `-vdev` option is specified, the devices are added to the virtual device list.

Note

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.

-tgt
When the `-tgt` option is specified, the devices are added to the target device list.

Note

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.

Move and copy devices in composite groups

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.

  Note

  The `move` and `moveall` commands fail if you attempt to move devices from a composite group containing storage groups. You must use the `symsg move` or `symsg moveall` commands to move devices in a storage 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.

  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.

You can assign a new name to a device being copied or moved.

Syntax

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]
```

Options

-sel_rdfg
When the **`-sel_rdfg`** option is specified, only SRDF devices belonging to that group number are moved or copied.

-r1
Limits the number of SRDF devices added to a composite group to R1s.

-r2
Limits the number of SRDF devices added to a composite group to R2s.

-r21
Limits the number of SRDF devices added to a composite group to R21s.

-noRDF

By specifying `-noRDF`, you can prohibit any SRDF devices from becoming members of the composite group.

Copy devices from a device group to a composite group

Description

Use the **`symcg dg2cg`** command to copy or add devices from an existing device group to a composite group. If the composite group does not exist, it is created. By default, all device lists from the device group are added to the composite group.

Note

During a device add operation, the command fails if the composite group already contains storage groups.

Syntax

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]
```

Options

-bcv
Adds only the BCVs to the composite group.

-vdev
Adds only the virtual devices to the composite group.

-hop2
Indicates 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.

-rtgt
Adds only the RTGTs to the composite group.

Remove a standard device from a composite group

Description

Use the **`symcg remove`** command to remove devices from an existing composite group. You can either remove devices individually, remove all devices, or remove all devices meeting the specified criteria.

Removing an consistency-enabled device does not disable the device. Use **`symcg disable`** to disable the composite group. If SRDF consistency is enabled and cannot be disabled, use **`-symforce`**.

Note

If the device is contained in a storage group, the operation will fail. Use the **symsg remove dev** or **symsg rmall** command to remove devices in a storage group.

Syntax

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]
```

Remove all standard devices from a composite group

Description

Use the **symcg rmall** command to remove all devices from an existing composite group.

Note

If the devices are contained in a storage group, the operation will fail. Use the **symsg remove dev** or **symsg rmall** command to remove devices in a storage group.

Syntax

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]
```

Options

-sel_rdfg SelRdfGrpNum

Indicates to remove only the devices in the specified group from the composite group

Rename a composite group

Description

Use the **symcg rename** command to rename a composite group.

Note

The command fails if you attempt to rename a logical device name of a device in a composite group containing storage groups. Also, you cannot rename an SRDF consistency composite group that is enabled.

Syntax

Use the following syntax:

```
symcg [-i Interval] [-c Count]
rename OldCGName NewCGName
```

Set SRDF group attributes

Description

Use the **symcg set** command to associate a logical name or an SRDF/Star recovery SRDF group number with an SRDF group to perform operations on all associated devices. If the **-rdfg** argument is not specified, then the action is applied to all SRDF groups.

Syntax

Use the following syntax:

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
   set -name [Name] | -recovery_rdfg GrpNum
    [-rdfg SymmID:GrpNum[,GrpNum,...]|all[,...] |
        name:RdfGroupName[,RdfGroupName]]
```

Composite group creation and output

Description

Use the **symcg list** command to obtain information about the composite groups visible to your host system.

Options

-inactive

In a GNS-enabled environment, returns 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 152

-novalid

Eliminates the validation of groups during the execution of the `list` command. The `Valid` column does not display in the output.

Syntax

Use the following syntax to list the host-visible composite groups:

```
symcg list
```

Examples

The following example shows how to create a non-consistent RDF1 composite group, associate devices with the group:

Create a non-consistent RDF1 composite group named `MyNewCG`.

```
symcg create MyNewCG -type rdf1
```

Add RDF1 devices to the composite group.

```
symcg -cg MyNewCG addall -devs CEE:CFD -sid 79
symcg -cg MyNewCG addall -devs 2:7 -sid 32
```

Associate RBCV devices with the 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
```

Associate RRBCV devices with the 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\
```

Sample output

As a result of the above command, the **`symcg list`** command returns the following output:

```
      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
```

Use the `symcg show` command to return additional details about a composite group.

Display composite group information

Syntax

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 (*). Display composite group information on page 144 explains how to set the environmental variable to display full group names.

Show composite group details

Description

Use the **`symcg show`** action to return additional details about a composite group.

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.

Show composite groups associated with a device group

Example

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 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
```

Perform control operations on composite group devices

Description

Use the **symcg -cg** command 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

Note

Virtual Pool (VP) compression (Enginuity 5876 only) on page 353 provides information on compressing and uncompressing thin devices.

- 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 or unset Geometry Compatible Mode (GCM)
- Set Optimized Read Miss mode (ORM)

Note

For arrays running HYPERMAX OS 5977, the Optimized Read Miss feature is not supported.

Syntax

Use the following command syntax to perform control operations on devices within a composite group:

```
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> ...]]
```

Restrictions

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.).
- 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.).

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
  ```

  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. 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.

- If SYMAPI_USE_GNS is set to ENABLE, all group information will be stored in a global database in VMAX or DMX arrays.
  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 157.

Note

If User Authorization is enabled, configuration and management of GNS requires a minimum role of StorageAdmin.

Automatic upgrade

When upgrading 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

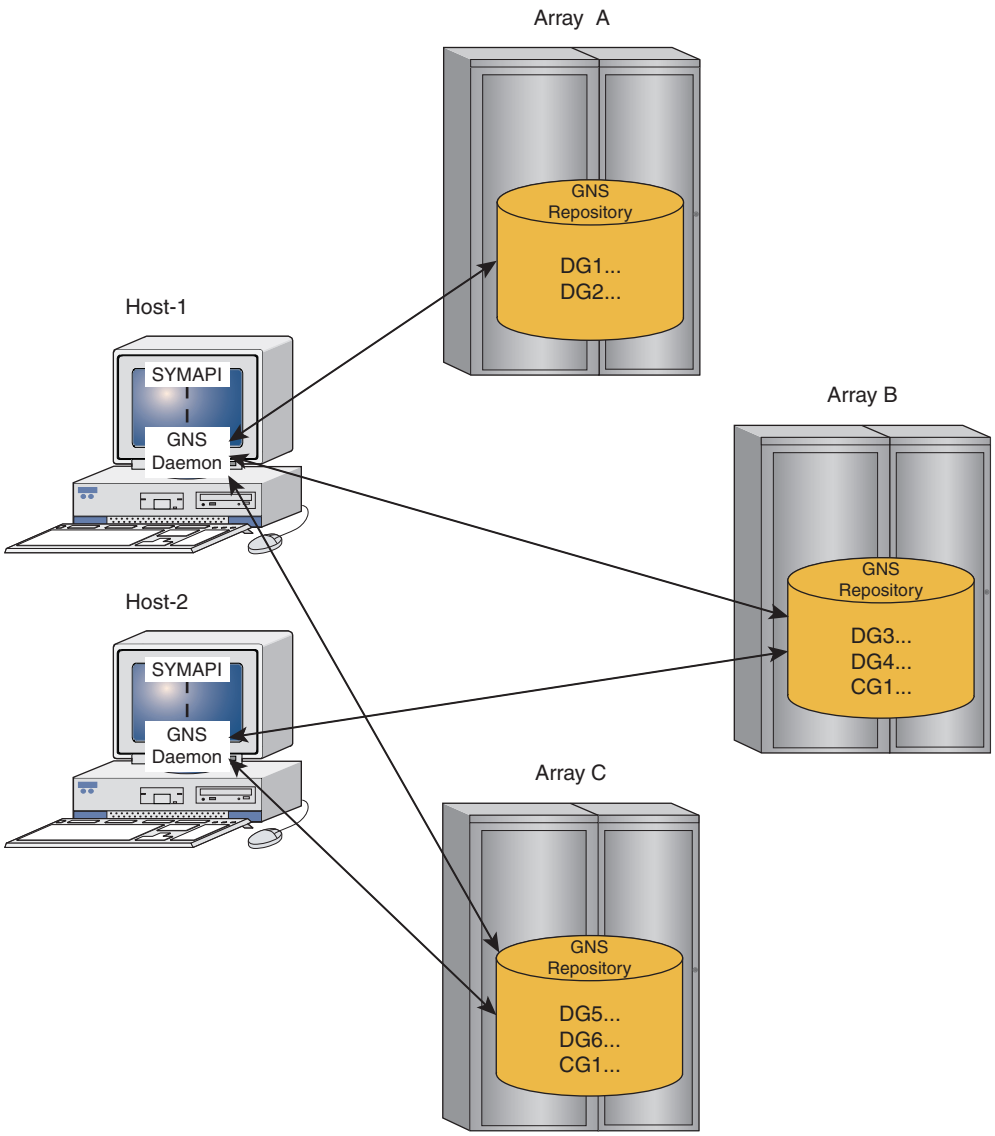
A *device group* is a user-defined object comprised of devices that belong to a single array and RA groups on that device. In a GNS-enabled environment, device group definitions are stored in the GNS repository of the array on which the devices within the device group reside and are visible to all hosts 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 5 on page 150, 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).

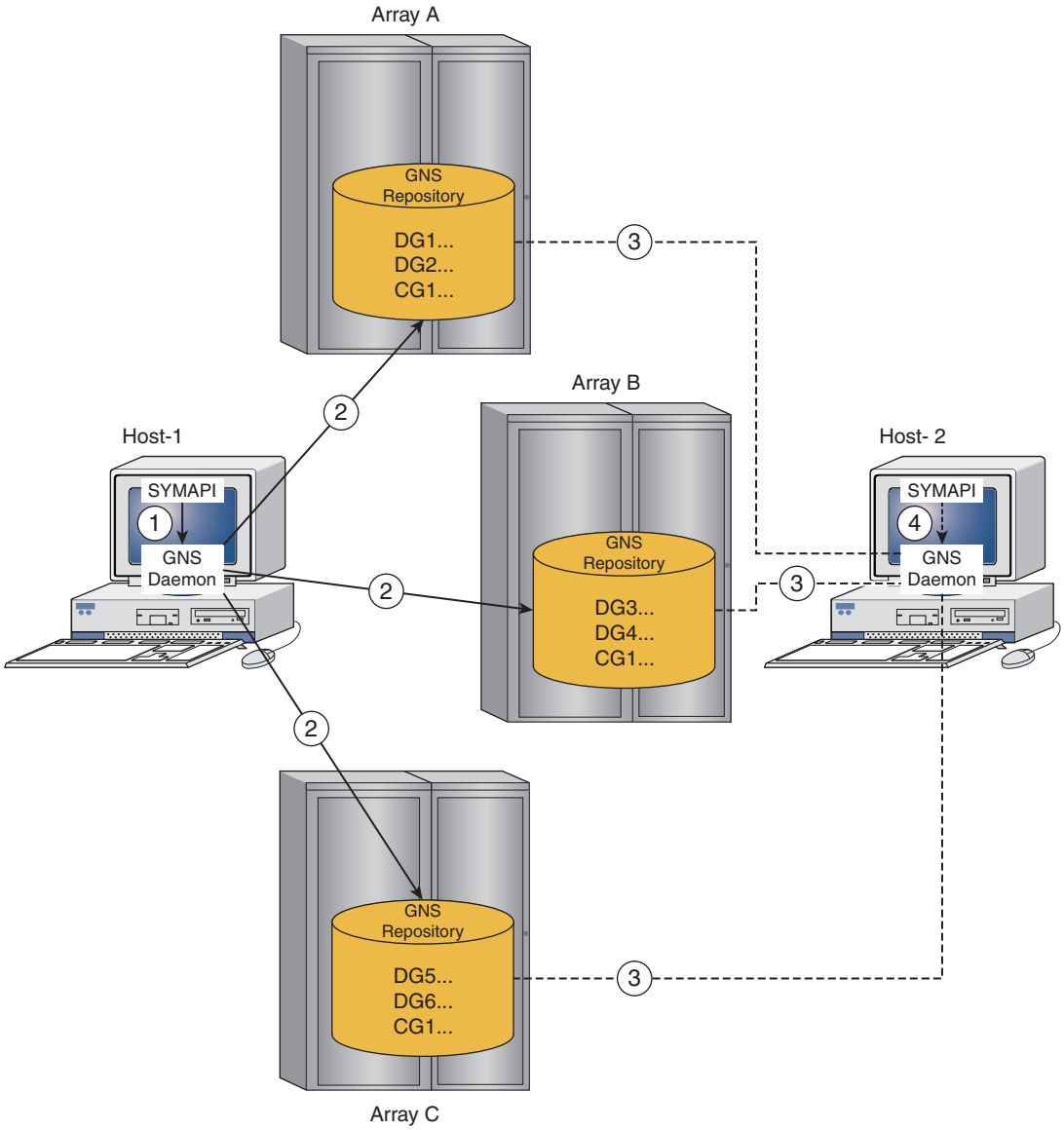
Figure 5 Host-visible GNS state



GNS updates to group definitions

When GNS is enabled, device group and composite group definitions are stored in the common GNS repository.

Figure 6 GNS across multiple hosts



For example, in Figure 6 on page 151, 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 Multi-Session 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 Create a composite group from devices in a storage group on page 136.

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 failover 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), failover 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 Remotely mirror device group definitions on page 159.

GNS behavior in client/server mode

In client/server mode, the SYMAPI_USE_GNS setting in the SYMAPI options file on the SYMAPI server 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, limited access to the group definitions held within a group database is provided while GNS is enabled.

GNS setup

Description

Use the `SYMAPI_USE_GNS` option in the SYMAPI options file to enable or disable GNS on each host.

Options

SYMCLI_USE_GNS

Enables or disables GNS on each host. Possible values are `ENABLE` or `DISABLE` (the default setting).

- 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.
- 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.

SYMCLI_DB_FILE

Determines the database to store groups.

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.

Note

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.

Examples

To enable GNS, enter:

```
SYMAPI_USE_GNS=ENABLE
```

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 through the options file, refer to the EMC Solutions Enabler CLI Command Reference. 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 see 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:

Table 9 Location of daemon authorization file

UNIX	`/var/symapi/config/daemon_users`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_users`

Note that non-root Solutions Enabler 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

Start the GNS daemon

There are three ways to start the GNS daemon:

- The daemon will be started automatically by theSolutions 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 current, since `storgnsd` uses the information stored in it to establish contact with your arrays.

- Start the daemon manually using the `stordaemon` command line utility as follows:

  ```
  stordaemon start storgnsd
  ```

- 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.

Restart the GNS daemon

If the GNS 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 `-autostart` 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.

The 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 157).

The restart functionality is provided as follows:

Table 10 Restarting the GNS daemon

Unix	By a dedicated watchdog daemon (`storwatchd`) which is started automatically as needed.
Windows	By the Service Control Manager.

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.

Manage 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.

Use the **stordaemon -h** command to display the command syntax.

Access groups in offline mode

Description

When accessing a non-default alternate group database file (for example, from a different host), there are two ways to specify the file.

- Specify a **SYMCLI_GROUP_DB** environmental variable to indicate the path of the group database file.
- Specify a **-grpfile** command line option for **symcg** and **symdg** commands.

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.

Syntax

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>]
. . .
```

View and release GNS daemon external locks

Description

The GNS daemon uses two external locks to maintain exclusive access to the GNS repository on each array: F0 and F1.

Syntax

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

Description

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

/var/symapi/log/storgnsd.log1

Windows

c:\Program Files\EMC\SYMAPI\log\storgnsd.log0

c:\Program Files\EMC\SYMAPI\log\storgnsd.log1

These two files are written in an alternating manner. When the active one becomes full, it is closed and the other one is truncated and made active.

Syntax

Use the following syntax to display the contents of the log file:

```
stordaemon showlog storgnsd -lines 200
```

GNS daemon options file

Description

Configuration options for the GNS daemon are contained within the daemon options file (**storgnsd**) located in the following directories:

Table 11 Locations of GNS configuration options

UNIX	`/var/symapi/config/daemon_options`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_options`

Syntax

Use the following syntax to specify options in the file:

```
storgnsd:OptName = OptValue
```

Options

OptName
The name of the option

OptValue
The new value

Modify option file values 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.

Modify option file values via the CLI

Description

Option file values can be set using the **stordaemon setoption** CLI utility, which programmatically makes changes to the **daemon_options** file.

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.

Syntax

The following is the command-line syntax for the **stordaemon setoption** option:

```
stordaemon setoption DaemonName
-name OptName=OptValue [-force]
```

Note

For information on the possible GNS daemon options file parameters, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*

Options

OptName
The name of the option

OptValue
The new value

Examples

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.

Modify option file values during runtime

Description

You can also set the daemon option value using the **stordaemon setvar** utility, which modifies the value during runtime. The command changes 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.

Syntax

The following is the syntax for the **stordaemon setvar** option:

```
Stordaemon  setvar  Daemonname  -name Var=Value
```

Options

Var
The name of the option

Value
The new value

Example

For example:

```
stordaemon setvar storgnsd -name autorestart=enable
```

Remotely mirror 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. Execute the **stordaemon reload** command to make the new option setting effective.

When enabled, GNS maintains 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.

As a result 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. 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.

Mirroring restrictions

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 non-standard SRDF devices

Modify mirrored group control

An SRDF group that has been created by the GNS remote mirror service 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. To change this behavior, set 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.

BCV and remote BCV device alias name swap

In the remote mirror name transformation, the following changes are made:

- 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*.
- The RBCV alias list becomes the BCV alias list in the remote mirror. The alias names that have a default value of RBCV*nnn* are automatically changed to BCV*nnn*.
- 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*.
- 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 you do not provide one. If you override the defaults and supply your own names that look like the above, they will still be transformed as described.

In the unlikely scenario that the you have 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.

The following example 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
```

Identify GNS groups

After configuring 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.

Identify device group definition

Description

For a device group, the output of the **symdg show** ***DgName*** command identifies whether a group definition is stored in GNS.

Example

```
symdg show MyRegDeviceGroup
```

Sample output

This sample output 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.

```
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 Locally-associated BCVs           :   20
    Number of Locally-associated VDEVs          :    0
. . .
```

Identify composite group definition

Description

For a composite group, the output of the **symcg show** ***CgName*** command identifies whether a group definition is stored in GNS.

Example

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, use the backup copy to restore the global database on arrays. For local groups, use the backup copy 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 backup to the GNS database

Description

To manually backup to the GNS database, use the **stordaemon** program.

Syntax
Use the following syntax for local backup:

```
stordaemon action storgnsd -cmd backup_db local
```

Use the following syntax for global backup:

```
stordaemon action storgnsd -cmd backup_db global
```

Manual restore from the GNS database

Description
To manually restore from the GNS database, use the **stordaemon** program.

Syntax
Use the following syntax for a local restore:

```
stordaemon action storgnsd -cmd restore_db local
```

Use the following syntax for a global restore:

```
stordaemon action storgnsd -cmd restore_db global changed_records:sid
stordaemon action storgnsd -cmd restore_db global all_records:sid
```

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.

Troubleshoot 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). However, duplicate-named groups may occur.

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. A subsequent consideration change could then allow the GNS daemon to see both groups.

Note

For information on renaming composite groups, see Rename a composite group on page 143.

Example
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. 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
```

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.

List GNS data about groups

The GNS **stordaemon list** command supports the following features:

- **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**
  Lists groups in a private GNS database determined by the environment variable SYMCLI_DB_FILE.

Show GNS data about groups

The GNS **stordaemon show** command supports the following features:

- **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**
  Shows 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**
  Shows 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 Tieringâ„¢ (FAST) operations.

- Fully Automated Storage Tiering (HYPERMAX OS 5977) 168
- Fully Automated Storage Tiering (Enginuity 5874 through 5876) 180

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 through 5876) on page 180 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* (SRP) by creating thin pools according to each individual disk technology, capacity and RAID type.

FAST technology moves the most active parts of 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 and helps reduce acquisition, power, cooling, and footprint costs. FAST factors 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 provides the ability to deliver variable performance levels through *Service Level Objectives*(SLO). Thin devices are added to *Storage Groups* and the storage group are assigned a specific SLO which sets performance expectations.

A SLO is the response time target for a storage group. The SLO sets the VMAX array with the desired response time target for a storage group. It automatically monitors and adapts to the workload in order to maintain the response time target. The Service Level Objective includes an optional workload type so it can be fine tuned to meet performance objectives.

There are five SLOs. For each one, except Optimized, the workload type, such as Online Transaction Processing (OLTP) or Decision Support System (DSS), is specified. The Optimized service level does not allow for setting the I/O type. A SLO cannot be modified, however a storage group can be assigned based on the required service level.

FAST monitors the storage groups performance relative to the SLO 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 (SRP) — one (default) FAST SRP is pre-configured on the array. This process is automatic and requires no setup.

Storage Resource Pool management

The `symconfigure set` command modifies Storage Resource Pool configuration. Configuration modifications include:

- set reserve capacity
- enable or disable SRDF/A DSE
- change SRP descripton

Modify storage resource pools

Syntax

To modify a Storage Resource Pool(SRP), use the following syntax:

```
symconfigure set srp <SRP Name>,
<[resv_cap = <n | NONE>]
[,rdfa_dse = <ENABLE | DISABLE]>
[,description = 'SRP
Description']>;
```

Options

resv_cap

A percentage of the capacity of the SRP 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 SRP 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

Existing TimeFinder snapshot sessions created on devices in the SRP are invalid if the free capacity of the SRP, as a percentage of the usable capacity, goes below the reserved capacity.

rdfa_dse

There is always one SRP assigned available for SRDF/A DSE allocations. Valid values are enable and disable. The default SRP for FBA emulation is used by default. Enabling rdfa_dse on a SRP will make that pool available for SRDF/A DSE allocations (and implicitly sets the currently assigned SRP to "disabled"). The maximum amount of storage from a SRP allowed for DSE is controlled by the system `wide dse_max_cap` setting, as described in the *EMC Solutions Enabler SRDF Family CLI User Guide.*

Note

This option is not allowed for SRPs with only external storage.

description

Modifies SRP description.

Examples

To set the existing SRP 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;"
```

To modify the description of an SRP, enter:

```
 symconfigure -sid 087 commit -cmd "set srp DEFAULT_SRP description =
'SRP with description modification';"

Execute a symconfigure operation for symmetrix '000197100087' (y/
[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change
session..............Established.
    Processing symmetrix 000197100087
    {
set srp DEFAULT_SRP description = 'SRP with description modification';
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute the `symconfigure set` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin.
- The specified SRP must exist.
- New parameters must be different than existing parameters.
- The target SRP for the operation cannot be the default SRP for FBA emulation, and the specified state cannot be "disabled".
- SRP description cannot exceed 127 characters (excluding null termination character). Valid values are "a - z", "A - Z", "0 - 9", underscore, hyphen"-", space, period".", and comma",".

Rename storage resource pools

Syntax

To rename a Storage Resource Pool (SRP), use the following syntax.

```
symconfigure rename srp <SRP Name> to <New SRP Name>;
```

Example

```
symconfigure –sid 087 commit -cmd "rename srp DEFAULT_SRP to
SRP_RENAMED;"

Execute a symconfigure operation for symmetrix '000197100087' (y/
[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change
session..............Established.
```

```
    Processing symmetrix 000197100087
    {
rename srp DEFAULT_SRP to SRP_RENAMED;
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute the `symconfigure set` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin.
- The specified SRP must exist.
- SRP name cannot exceed 32 characters. Valid characters are only alphanumeric characters, hyphens "-", and underscores "_", with leading hyphen or underscore not allowed.
- In a single command file that includes "set srp" and "rename srp" commands, the following rules apply:
  - Allows using the old SRP name for "set srp" commands and renaming the SRP as the last command.
  - Allows renaming the SRP as the first command and using the new name for the subsequent "set srp" commands.
  - Does not allow using the old SRP name for some commands, changing the SRP name, and then using the new SRP name for subsequent commands.

FAST information reporting

FAST reporting lists the Service Level Objectives, storage groups, and Storage Resource Pools configured on the array, and also generates reports detailing the demand that storage groups or Service Level Objectives are placing on the Storage Resource Pools.

List Service Level Objectives

Description

The `symcfg list` and `symcfg show` commands report Service Level Objectives (SLO)s and available workloads.

Options

-all
: Displays only the SLOs that are allowed for the specific VMAX array.

-v
: Provides a verbose list of SLOs and workloads with additional descriptions.

Example

To list SLO details on array 230, enter:

```
symcfg -sid 230 list -slo -detail -all
```

Sample Output

```
           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
```

Show specific Service Level Objective

Description

The `symcfg show -slo` command displays a specified Service Level Objective (SLO).

Example

To show information about the Platinum SLO for array 230, enter:

```
symcfg -sid 230 show -slo Platinum
```

Sample Output

```
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
  }
```

List storage groups

Note

Storage groups on page 100 explains how to create storage groups and how to add and remove devices from a group.

Description

The `symsg list -detail` command shows details for all storage groups, including Service Level Objective (SLO), workload, and Storage Resource Pool (SRP) configured for each storage group.

Options

-by slo

Sorts storage group information by SLO.

-by srp

Sorts storage group information by SRP.

Example

```
symsg -sid 230 list -detail
```

Sample output

Note

Starting with Solutions Enabler 8.0.1 the `FAST R` column for listing RDF coordination (enabled or disabled) has been removed from the output.

- If the SLO or SRP name length exceeds 24 characters output display truncates the name to 23 characters and displays "*" for the 24th character.
- When the (F)ast flag is set on an array running HYPERMAX OS 5977, it indicates that the storage group has a SLO or a SRP set and it is FAST-managed.
- 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.

```
S T O R A G E  G R O U P S
Symmetrix ID:        000197100230
                       Flags Number   Child
Storage Group Name     EFMSL Devices  SGs     SLO Name   Workload    SRP Name
-------------------------------------  ---------- --------   --------------------
AcctPay                FX...     250      0    Gold       OLTP      Primary_SRP
Customer               AX...      38      0    Optimized  <none>    Primary_SRP
Ordering               MX...      19      0    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
```

List Storage Resource Pools

Example

The **`symcfg list -srp`** command displays all Storage Resource Pools (SRP)s configured on an array.

```
symcfg -sid 087 list -srp -detail
```

Sample output

Output with **`-detail`** option (reports capacity metrics) :

```
          STORAGE RESOURCE POOLS

Symmetrix ID  : 000197100087
                                                    C A P A C I T Y
```

```
-------------------------------- ---
-------------------------------------
                                 Flg     Useable  Allocated
Free Subs
Name                             DR         (GB)        (GB)
(GB)   (%)
-------------------------------- --- ---------- ---------- ----------
----
Default_SRP                      FX     78848.0    42880.0
35968.0  125
Test_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
```

Output descriptions:

- Usable — indicates the usable capacity of all the disk groups in the SRP.
- Allocated — indicates the sum of the device allocations, snapshot allocations, and SRDF/A DSE allocations on the SRP.
- 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 SRP.

Note

The capacity values are listed in gigabytes by default. Other options are -MB (megabytes) or -TB (terabytes).

- Default SRP — indicates if the SRP is the default for FBA devices.
- RDFA DSE — indicates if the SRP usable for SRDF/A DSE operations.

Output without **-detail** option (no capacity metrics):

```
                            STORAGE RESOURCE POOLS

Symmetrix ID  : 000197100087
------------------------ --- ----------------------------------------
                         Flg
Name                     DR  Description
------------------------ --- ----------------------------------------
DEFAULT_SRP              FX  The Default SRP that will be used for all
                                 applications
TEST_SRP                 ..  SRP for testing

Legend:
  Flags:
   (D)efault SRP : F = FBA Default, . = N/A
   (R)DFA DSE    : X = Usable, . = Not Used
```

List Storage Resouce Pools with verbose option

Example

To display a verbose list of all SRPs in array 230, enter:

```
symcfg -sid 230 list -srp -v -gb
```

Sample output

```
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):
  {
   ----------------------------------------------
                                          Useable
                                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
Useable 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
Useable 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
  }
```

Output descriptions:

- Default SRP — indicates if the SRP is the default pool for FBA devices.
- Usable Capacity — sum of the useable capacity of all the disk groups in the SRP.
- Allocated Capacity — sum of the device allocations, the snapshot allocations, and the SRDF DSE allocations on the SRP.
- Free Capacity — difference between the useable and the allocated capacity.
- Subscribed Capacity — sum of the sizes of all the thin devices subscribed against the SRP.
- Subscribed Capacity percentage — percentage of the Usable Capacity used by thin devices.
- Reserved Capacity — percentage of the Useable Capacity reserved for non-snapshot activities.

Note

Existing TimeFinder snapshot sessions created on devices in the SRP may become invalid if the Free Capacity of the SRP, as a percentage of the Usable Capacity, goes below the Reserved Capacity.

- Useable by RDFA DSE — indicates if the SRP is usable for SRDF/A DSE operations.
- Disk Groups — lists Tte disk groups configured to the SRP. The list of disk groups is sorted by disk group number.
- Available SLOs — lists the SLOs that are available for this SRP.

Show specific Storage Resource Pool

The `symcfg show -srp` command displays a specified Storage Resource Pool (SRP).

Options

-detail
Shows detailed information about a specific SRP.

Example

To show detailed information about the pool named Primary_SRP, enter:

```
symcfg -sid 230 show -srp Primary_SRP -detail -gb
```

Sample output

```
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)   Product
Name                                             Array ID
   --- -------------------- ---- ----- ----------
-------------------- --------------------------------
     1 DiskGroup_1          SATA  7200    20480.0
VMAX                   00019710230
     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
  }
                            Flgs Speed   Capacity
   #   Name                 LTS (rpm)       (GB)  Product
Name         Array ID
   --- -------------------- ---- ----- ----------
-------------------- --------------------------------
     1 EXTERNAL_DG_001      X-N  10000     1409.1
XtremIO             000197300005
                                        ----------
   Total                                    1409.1
```

Storage Resource Pool demand reporting

The `symcfg list -srp -demand` command reports the subscribed and the allocated demand, from Service Level Objectives and Storage Groups, placed on the Storage Resource Pools (SRP)s.

Report Service Level Objective demand on Storage Resource Pools

Options

-type slo
: Reports the demand placed by Service Level Objectives (SLO)s on the Storage Resource Pools (SRP)s. 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 SLO.

-details
: Reports the SRP details.

Example

To report detailed SLO demand on the SRPs, enter:

```
symcfg -sid 230 list -srp -demand -type slo -detail -GB
```

Sample output

```
                        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
```

Report storage group demand on Storage Resource Pool

Options

-type sg

Reports the demand placed by storage groups on the Storage Resource Pools (SRP)s.

Example

To report storage group demand on the SRPs, enter:

```
symcfg -sid 230 list -srp -demand -type sg -GB
```

Sample output

```
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
```

List Storage Resource Pools for thin devices

Description

For thin devices in an array, the `symcfg list -tdev -srp` command lists:

- configured size
- associated Storage Resource Pools (SRP)s
- device allocations
- snapshots allocated to SRP
- a range of devices or specific storage group (if device range or storage group name is provided in the command syntax)

Example

To list a range of thin devices associated SRPs, enter:

```
symcfg -sid 584 list -tdev -devs 13e9:1400 -srp
```

Sample output

```
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 through 5876)

Note

Fully Automated Storage Tiering (HYPERMAX OS 5977) on page 168 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). FAST and FAST VP comparison on page 181 shows the differences.

FAST implementation provides the following benefits:

- Reduces acquisition costs by maximizing utilization of less expensive drives (SATA) for infrequently used data
- Improves performance by optimizing data placement for high access requirements data
- Reduces overall operating costs and simplify management by consolidating systems, using fewer drives, and reducing power, cooling, and floor space requirements
- Reduces the total cost of ownership by optimizing the performance of a tiered system.

Available tools to monitor a FAST configuration, include:

- 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 216.
- Many **`list`** and **`show`** commands are available to examine the details of the FAST configuration. These displays are shown in List FAST tiers on page 188, List FAST DP tiers on page 190, and List FAST DP tiers on page 190. 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.

FAST and FAST VP comparison

Table 12 FAST version comparisons (Enginuity 5874 through 5876)

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

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 Manage FAST tiers on page 182.

- Define a FAST policy — Policy configuration rules and restrictions are explained in FAST policy management on page 191.
- Define storage groups — Storage group configuration rules and restrictions are explained in Manage storage groups on page 387.

FAST configuration restrictions

The following configuration restrictions 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 Time windows overview on page 606
- Control parameters define the numbers of and types of devices, the mode of operation, the thresholds for data movement, and the analysis time period.

Manage FAST 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 are 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
- Creating virtual pool tiers
- Adding disk groups to a tier
- Removing disk groups from a tier
- Modifying a VP tier
- Adding thin pools to a tier
- Removing thin pools from a tier
- Renaming tiers
- Deleting tiers
- Listing tier information
- Showing tiers

FAST tier restrictions

The following restrictions apply to tiers:

- Maximum tier name size is 23 characters and includes alphanumeric characters, hyphens (-), and underscores (_). 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), and are not case sensitive.
- The maximum number of tiers allowed on an array is 256, including 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 must match the tier technology. Do not specify both the disk group name and the disk group ID in the same command.
- The protection types 3+1, and 7+1 are only valid when used with `–tgt_raid5` option, and protection type 6 + 2 and 14+2 are only valid when used with `–tgt_raid6` option .
- An error returns on the `symtier create` command, if there are no matching disk groups to the tier technology.
- An error returns on the `symtier create` command, 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 new DP tier cannot 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, both static VP tiers and dynamic DP tiers of the same technology type are allowed, as well as 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 VP tier requires pools that are thin pools, contains DATA devices, and are enabled.
- The thin pool and thin tier technology type must match, so 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.
- The maximum number of thin pools in a VP tier is 4.
- A thin pool can only be included in one VP tier.

Create FAST disk group tiers

Description

When creating a VMAX array tier for disk group provisioning, specify a tier name, the protection type (RAID 1, RAID 5, RAID 6), the technology (EDF, FC, SATA), and how the disk groups are added to the tier. There are two ways disk groups are 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 expands to accommodate any newly added disk groups.
- Static — A static tier is manually populated with disk group names or disk group IDs. Both disk group name and disk group ID cannot be specified using the same command. If the *DiskGroupID* is not specified, an empty tier is created.

When the FAST controller moves a device to a tier, it changes the device's protection type to match the tier's protection type.

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 decides whether to move one or more devices to this tier. After the devices are moved to this tier, the devices are in disk group 1 and their protection type is RAID 1.

Syntax

To create a tier, use the following syntax:

```
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...]>]]
```

Options

The **-interval** and **-count** options are used with the **symtier create** command, and the command retries if the SYMCLI fails to get an exclusive lock on the SYMAPI database.

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
```

Create FAST virtual pool tiers

Description

When creating a virtual pool (VP) tier, specify a tier name, the protection type (RAID 1, RAID 5, RAID 6), the technology (EDF, FC, SATA, external), and thin pools names. As 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.

Syntax

Note

Since thin tiers are static only, do not use the -inc_type option in the syntax.

To create a thin tier, use the following syntax:

```
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...]>]
```

Options

-technology
Specifies the technology type for the tier. SATA is the default if `-technology` is not specified.

-external
Indicates the tier is external to the array.

Examples

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 VP tier restrictions

The following restrictions apply to creating external virtual tiers with Federated Tiered Storage (FTS):

- External tiers are only allowed with Enginuity 5876 or higher.
- The **`-tgt_unprotected`** option is only valid for external tiers, and 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.

Add disk groups to a FAST tier

Syntax

To add disk groups to a tier, use the following syntax:

Note

A disk group can only be added to a DP tier if the tier type is `static`.

```
symtier -sid SymmID [-i Interval] [-c Count]add -tier_name TierName [-
propagate]
-dsk_grp <DiskGroupID[,DiskGroupID...]> |
<name:DiskGroupName[,DiskGroupName...]>
```

Options

-propagate
Tiers in a VMAX array cannot partially overlap, therefore this option adds the new disk group to all tiers that overlap with the tier being modified. This option only works when adding a disk group that does not participate in any tier.

Examples

To add disk group 3 to an overlapping tier (TierA overlaps with TierB), enter:

```
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 has only disk group 1, and TierB has only disk group 2, Solutions Enabler does not provide a way to modify TierA and TierB to contain disk group 1 and 2. These tiers must be deleted and recreated with disk groups 1 and 2.

Remove disk groups from a FAST tier

Syntax

Note

A disk group can be removed from a DP tier if the tier type is **static**.

To remove a disk group form a tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count] remove -tier_name
TierName [-propagate]
-dsk_grp <DiskGroupID[,DiskGroupID...]> |
<name:DiskGroupName[,DiskGroupName...]>
```

Options

-propagate
This option removes the disk group from all tiers that overlap.

Examples

For example, to remove disk group **2** from **PrimeTier**, enter:

```
symtier -sid 207 remove -dsk_grp 2 -tier_name PrimeTier
```

Modifying a FAST VP tier

Syntax

To modify a FAST VP tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count] modify -tier_name
TierName
-technology <EFD | FC | SATA>
```

Options

-technology
Specifies the tier type when modifying an external tier.

Add thin pools to a FAST tier

Syntax

To add a thin pool to a VP tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count] add -tier_name TierName -
pool <PoolName[,PoolName...]>
```

Examples

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
```

Remove thin pools from a FAST VP tier

Syntax

To remove a thin pool from a FAST VP tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count] remove -tier_name
TierName -pool <PoolName[,PoolName...]>
```

Removing all thin pools from a FAST VP tier if the tier is part of a FAST policy is not allowed. Removing 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 is not allowed.

Rename FAST tiers

Syntax

To rename a FAST tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count]
rename -tier_name TierName -name NewTierName
```

Note

New FAST tier names must adhere to all . on page 182

Examples

To rename the tier **Primary** to **PrimeTier**, enter:

```
symtier -sid 207 rename -tier_name Primary -name PrimeTier
```

Delete FAST tiers

Syntax

Note

Before deleting a tier, verify that it is not associated with any FAST policy.

To delete a tier, use the following syntax:

```
symtier -sid SymmID [-i Interval] [-c Count]
delete -tier_name TierName [-force]
```

Options

-force
Required to delete a non-empty static tier.

Examples

To delete a tier named **DBTier**, enter:

```
symtier -sid 207 delete -tier_name DBTier
```

List FAST tiers

Description

The **symtier list** command gives a summary of all the disk group or virtual pool tiers in the array. Use the `symtier show` command for information about individual tiers.

Syntax

```
symtier [-sid SymmID] [-v] [-offline]
    list [-dp | -vp [-ckd] [-fba]]
         [-technology <EFD | FC | SATA> | -external]
symtier show -sid SymmID [-offline]
```

Options

-dp or -vp
Used without the `-v` option, and lists only disk group provisioned (`-dp`) or only virtual provisioned (`-vp`) tiers . All tiers display if neither option is specified.

-offline
Prior to using this option, populate the SYMAPI database with tier information using the `symcfg sync -fast` command.

Examples

To list all the tiers on VMAX array `432`, enter:

```
symtier list -sid 432
```

Sample output

Note

The `emul` (emulation type column) lists the emulation type for FAST VP tiers. For FAST DP tiers and empty FAST VP tiers this column lists `N/A`.

```
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
```

List FAST VP tiers

Example

To list all the thin tiers (`-vp`) on VMAX array **`432`**, enter:

```
symtier list -vp -sid 432
```

Sample output

```
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
```

Output description:

- `Tier Name` — Name of the tier.
- `Location` — Internal or external tier.
- `Tech` — Technology of the tier.
- `Target Protection` — Tier protection type.
- `Emul` — Device emulation type.
- `Flgs ID` — Static or dynamic tier, and enabled or disabled 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 is counted, including DATA devices not enabled; therefore `Used` may be greater than `Enabled`.

List FAST DP tiers

Example

To list all the disk group tiers (`-dp`) on VMAX array **`234`**, enter:

```
symtier -sid 234 list -dp
```

Sample output

```
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
```

Output description:

- `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 is 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.

  Note

  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 is 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 are 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 is 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.

Show FAST tiers

Examples

To show the tier AS400_TIER, enter:

```
symtier -sid 432 show -tier_name AS400_TIER
```

Sample output

```
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
```

FAST policy management

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.

Policy restrictions

The following restrictions apply to FAST policies:

- Policy names must be unique and cannot exceed 32 characters. Policy names are case insensitive. For example, **`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.
- Up to 256 policies are allowed on an array.
- There is a maximum of up to four VP tiers in a storage policy, and each tier must be unique, there no overlapping disk groups are allowed.
- Empty tiers are not allowed..
- 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 added to a policy determines the type of tier for the policy. The fourth tier in a FAST VP policy must 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 values for the tier (*MaxSgPercent*) are greater than 0 and less than or equal to 100. All other values are rejected.

Creating a FAST policy

Syntax

To create FAST policies, use the following syntax:

Note

A new FAST policy, requires the VMAX array ID, a policy name, at least one tier name, and the upper limit (**MaxSgPercent**) of the tier that the storage group can occupy. If the upper limit is not specified the default is 100.

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
   create -name FastPolicyName [-tier_name TierName
         [-max_sg_percent MaxSgPercent]]
```

Options

-emulation

This option is available only for arrays running Enginuity 5876 or higher. If the emulation type is not provided, the default emulation is FBA. Mixed emulation VP types are not allowed in the same policy.

MaxSgPercent

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. Therefore, 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. The FAST controller uses this capacity percent as an upper limit and does not exceed the limit. If Tier 1 assigned capacity is 50% (100GB per tier), the FAST controller still only places at maximum two devices on the Tier 1 since adding one more device exceeds the upper limit of 50%.

Note

The **-MaxSgPercent**is 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 are counted, not logical device capacity.

Examples

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.

Add a tier to a FAST policy

Syntax

To add a tier to policy, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName  add -tier_name TierName-
max_sg_percent MaxSgPercent
```

Example

To add the tier **ArchiveDBTier** to the policy **DBPolicy** and specify that the tier contains 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
```

Restrictions

The following restrictions apply when adding a tier to a FAST policy:

- An empty tier cannot be added; the maximum tiers for each policy is three.
- Adding a new tier cannot result in two tiers in the same policy sharing a disk group.
- When adding a VP tier to a FAST policy, the policy must be empty or contain only VP tiers.
- VP tiers in a policy must have the same emulation.
- A given VP tier may be included in multiple FAST policies.

Modify limit for FAST policy tier

Syntax

To modify the limit of the tier capacity available to a storage group, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count] -fp_name
FastPolicyName
modify -tier_name TierName -max_sg_percent MaxSgPercent
```

Examples

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
```

For above example, no more than **70**% of the storage group associated with this policy will reside on **PrimeDBTier**. If the policy is associated with a storage group, the sum of the limits must be equal to or greater than 100%.

Remove tier from a FAST policy

Syntax

To remove a tier from a policy, use the following syntax:

Note

Before removing a tier from a policy, verify that 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. Do not remove a VP tier from a policy if thin devices in the associated storage group are bound to pools in the tier.

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
-fp_name FastPolicyName   remove -tier_name TierName
```

Examples

To remove **ArchiveTier** from the **FinanceData** policy on VMAX array **207**, enter:

```
symfast -fp -sid 207 -fp_name FinanceData remove -tier_name
ArchiveTier
```

Rename FAST policy

Syntax

To rename a FAST policy, use the following syntax:

Note

When renaming a policy, verify that the new name does not already exist.

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
-fp_name FastPolicyName   rename -name NewFastPolicyName
```

Examples

To rename **DBPolicy** to **OraDBPolicy** on VMAX array **207**, enter:

```
symfast -fp -sid 207 rename -fp_name DBPolicy -name OraDBPolicy
```

Delete a FAST policy

Syntax

To delete a FAST policy, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
-fp_name FastPolicyNamedelete -fp_name FastPolicyName [-force]
```

Examples

To delete FAST policy **HRPolicy** on VMAX array **207**, enter:

```
symfast -fp -sid 207 delete -fp_name HRPolicy
```

Restrictions

The following restrictions apply to deleting FAST policies:

- The **-force** flag must be used to delete a policy with tiers.
- 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, then delete the policy.

List FAST policies

Syntax

To list FAST policies, use the following syntax:

```
symfast [-sid SymmID] [-offline] [-v] list -fp [-v] [-dp | -vp [-ckd
| -fba]]
```

Note

Prior to using the `-offline` option, the SYMAPI database must be populated with the FAST information using the `symcfg sync -fast` command.

Options

-dp, -vp
Filters the output for FAST DP (`-dp`) or FAST VP (`-vp`). If neither option is used, all policies display.

-ckd, -fba
Filters the output for CKD (`-ckd`) or FBA (`-fba`) devices.

Examples

To list FAST policies on array 601, enter:

```
symfast -sid 601 list -fp -v
```

Sample output

Note

When listing a FAST VP policy with the -v (verbose) option, a `Flgs` (flags column) indicates if the tier is compression capable.

```
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
```

Show FAST a policy

Syntax

To show a FAST policy, use the following syntax:

```
symfast [-sid SymmID] [-offline] show -fp_name FastPolicyName
```

Note

Prior to using the **-offline** option, the SYMAPI database must be populated with the FAST information using the command **symcfg sync -fast**

Examples

To show the FAST policies for VMAX array, enter:

```
symfast -sid 601 show -fp_name test_policy
```

Sample output

```
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
```

FAST-related storage groups

Storage groups are a collection of devices on the VMAX array used for masking/mapping, Virtual LUN Technology, and FAST operations. This section explains the FAST-related storage group options.

Note

esd_r_se_seam_adding_cascaded_storage_groups_(enginuity_5876_and_hypermax_os_5977_only
page 111 explains how to create a storage group and how to add and remove devices from a group.

FAST-related storage group and device restrictions

The following restrictions apply to storage groups that are associated with FAST policies:

- A storage group is allowed to 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 maximum storage group associations to policies per array is 8192. 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 is allowed to be associated with only one policy. An attempt to associate a storage group with a device that is already associated to a policy is 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 must 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 is used to determine compliance. When the FAST policy contains DP tiers, FAST will only operate on the standard devices, and only the capacity from standard devices is used to determine compliance.
  All devices in the storage group are considered under FAST control, however, either the thin devices or the standard devices are selected for FAST management, not both.
- The following devices are not moveable and cannot be used in FAST-related storage groups:
  - 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 a FAST policy contains VP tiers, none of the thin devices can be bound to a pool outside of the policy.
- A FAST policy can contain multiple VP tiers with the same technology type. However, FAST VP only performs *compliance* movements between such tiers. There is 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.

Associate storage group to a policy

Description

This operation associates a storage group to an existing FAST policy and assigns a priority value (`-priority`) 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.

An association is successful when the storage group configuration rules are followed and there is enough capacity allocated in the tiers that belong to the FAST policy. The association is not blocked if the tiers do not have sufficient capacity, but the storage group is out of compliance with its policy. After the storage group is associated with the policy, the FAST controller actively monitors the storage group and suggests or runs optimizations to improve performance.

Syntax

To associate a storage group with a FAST policy, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count] -fp_name
FastPolicyName
associate -sg SgName [-priority PriorityValue]
```

Note

Associating a parent storage group with a FAST policy is not allowed.

Options

-priority

Priority values are 1, 2, or 3, with priority 1 as the highest value and 3 the lowest value. The default priority value is 2.

Examples

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
```

Set FAST SRDF coordination

Description

For Enginuity 5876 the `-rdf_coordination enable|disable` option is available to coordinate FAST VP actions on local and remote arrays. This instructs FAST to consider the R1 device statistics into the move decisions that are made on the R2 device. RDF coordination can be set even if there are no SRDF devices in the storage group. SRDF coordination requires that both the R1 and the R2 devices reside on an array running Enginuity 5876.

For Enginuity versions lower than 5876, there is no coordination of FAST VP actions (moves) between the local and remote VMAX arrays. Both sets of devices are managed individually, which can impact RDF performance.

Note

For HYPERMAX OS 5977, SRDF/FAST coordination is enabled by default.

Syntax

To associate a storage group with a FAST policy and set RDF coordination, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
-fp_name FastPolicyName associate -sg SgName [-priority PriorityValue]
[-rdf_coordination <ENABLE | DISABLE>]
```

Reassociate a storage group to a new policy

Description

When a storage group is reassociated to another policy, all the current attributes (such as priority or RDF coordination), set on the original association, automatically propagate to the new association.

Syntax

To reassociate a storage group to another policy, use the following syntax:

```
symfast -sid SymmID [-i Interval] [-c Count]
-sg_name SgNamereassociate -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 results 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.

Disassociate storage group from a policy

Description

When a storage group is disassociated from a FAST policy, the FAST controller no longer tries to optimize the performance of that storage group. The disassociate operation occurs, even if the FAST controller is currently moving devices in the storage group. Disassociating a storage group containing thin devices removes those devices in the storage group from FAST control. Thin device data remains at it's current location, even if it is spread across multiple pools, and is not moved back to the bound pool.

Syntax

To disassociate a storage group from a policy, use the following syntax:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
-fp_name FastPolicyNamedisassociate -sg SgName
```

Examples

To disassociate storage group **OraSales** from the FAST policy **DBPolicy**, enter:

```
symfast -sid 207 disassociate -sg OraSales -fp_name DBPolicy
```

Modify a storage group

Description

For Enginuity 5876 only, use the `symfast modify` command storage group to modify (priority and RDF coordination). Setting the priority is allowed only if a storage group is associated with a FAST policy, and setting and RDF coordination is only allowed for FAST VP. Priority values are 1, 2, or 3, with priority 1 as the highest value and 3 the lowest value. The default priority value is 2.

Syntax

To change a storage group priority or the SRDF coordination attribute, use the following syntax:

```
symfast -sid SymmID [-i Interval] [-c Count] -fp_name FastPolicyName
   modify -sg SgName [-priority PriorityValue][-rdf_coordination
<ENABLE | DISABLE>]
```

Examples

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
```

List storage group FAST associations

Description

Use the `symfast list -association` command to display the storage groups, their priority, their associated policies, and whether the RDF coordination attribute is set.

Note

If using the `-offline` option, the SYMAPI database must be populated with the storage group, tier, and FAST data using the `symcfg sync -fast` command.

The output is sorted alphabetically by storage group name.

Examples

To display the storage group associations on VMAX array `432`, enter:

```
symfast list -sid 432 -association
```

Sample output

```
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
```

Show storage group FAST associations

Description

The `symfast show -association` command to displays information about associations between storage groups and FAST policies containing standard or thin tiers. By default, only the devices managed by FAST display, either standard or thin devices, based on the policy. Gatekeeper devices are filtered out.

Syntax

```
symfast [-sid SymmID] [-offline]show  -sg SgName [-all] [-v]
```

Options

-all

Shows all devices in the storage group, regardless of the associated policy.

-v

The verbose option. 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.

Examples

To show the associations for storage group hr_sg on VMAX array **432**, enter:

```
symfast show -sid 432 -association -sg hr_sg
```

Sample output

```
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

For RDF coordination, shows as:

- `Enabled` or `Disabled` for Enginuity 5876.
- `Disabled` for Enginuity lower than 5876.
- `N/A` for FAST DP.

Note

The order of the associated tiers display highest to lowest, meaning the highest performing tier is listed on top and the lowest performing tier is listed on the bottom.

FAST controller management

The FAST controller is the intelligence of the FAST software. The FAST controller runs its algorithms continuously in the background, checking for FAST policies to initiate a performance improvement for the applications managed by those policies.

The FAST controller has two states — **Enabled** and **Disabled**. If disabled, there is no data movement between tiers based on the FAST policies. If enabled, there is data movement between tiers based on the FAST policies. By default, the FAST controller ships in a disabled state. To change the state, use the `symfast` command, to enable or disable the FAST controller.

With the FAST controller:

- EFD is the highest performing tier, FC as the second highest, and SATA the slowest performing tier.
- external tiers are considered slower than SATA tiers.
- it tries to place the busiest volumes on EFD, and ensures that it does not exceed the upper limit set by the FAST policy.

Table 13 Differences between FAST and FAST VP

FAST	FAST VP
• The FAST controller, which resides on the Service Processor, is solely responsible for FAST data movements. • The FAST controller gathers and analyzes device statistics then issues explicit movement instructions to the array.	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. • The FAST controller issues explicit movement instructions for thin devices, but only for compliance movements. • The FAST controller sends to Enginuity a set of partial instructions, for each storage group under FAST control, called a movement policy.

Table 13 Differences between FAST and FAST VP

FAST	FAST VP
	• Enginuity interprets these instructions and independently executes performance movements for thin devices, based on the device statistics. • **Note** In the event of a loss of the service processor, performance statistics collection continues. Performance data movements also continue for a short period of time, but will eventually stop, until the service processor is recovered.

The `symfast` command provides the following actions for managing FAST:

- Enable and disable FAST
- Display FAST state
- Manage time windows
- Set FAST control parameters
- Display FAST control parameters
- Display FAST plans
- Approve FAST plans
- Decline FAST plans

Enable FAST

Syntax

To enable the FAST controller, use the following syntax:

Note

Solutions Enabler supports FAST controls for local and remote arrays running Enginuity 5876.

```
symfast -sid SymmID [-i Interval]
[-c Count] [-noprompt] enable [-dp | -vp] disable [-dp | -vp]
```

Options

-dp
Enables FAST disk group provisioning.

-vp
Enables FAST VP (virtual pool) provisioning.

Note

If neither option is supplied, both FAST and FAST VP are enabled.

Examples

For example, to enable the FAST controller for disk group provisioning, enter:

```
symfast -sid 207 enable -dp
```

Disable FAST

Syntax

To disable the FAST controller, use the following syntax:

Note

Solutions Enabler supports FAST controls for local and remote arrays running Enginuity 5876.

```
symfast -sid SymmID [-i Interval] [-c Count] [-noprompt] disable [-dp
| -vp]
```

Options

-dp
Disables FAST disk group provisioning.

-vp
Disables FAST VP (virtual pool) provisioning.

Note

If neither option is supplied, both FAST and FAST VP are disabled.

Examples

To disable the FAST VP controller, enter:

```
symfast -sid 207 disable -vp
```

Restrictions

- Depending on the state of the FAST controller, the **disable** action may 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. If there is a FAST standard plan executing, the current group completes before the FAST controller is disabled. If the FAST controller is executing thin data movements for compliance, those data movements complete before the FAST controller is disabled.
- If the disable command fails, for example, because the FAST controller is unresponsive or DisabledWithError, performance movement may continue for thin devices. In this case, the only way to stop the thin data movement is to change the **-thin_data_move_mode** control parameter to NONE. Refer to Set FAST control parameters on page 207

Display FAST state

Description

To verify the state of the FAST controller, use the `list -state` command, specifying the `-vp` or `-dp` options. If `-vp` or `-dp` are not specified, both options display.

Options

-dp

Displays FAST disk group provisioning.

-vp

Displays FAST VP (virtual pool) provisioning.

Note

If neither option is supplied, both FAST and FAST VP display.

Examples

Using `-dp` option:

```
symfast -sid 234 list -state -dp
```

Using `-vp` option:

```
symfast -sid 234 list -state -vp
```

Sample output

Using `-dp` option:

```
Symmetrix ID: 000194900234
    FAST State              : Enabled
    Degraded Reason         : N/A
    Error Message           : N/A
    Current Activities      : Idle
```

Using `-vp` option:

```
Symmetrix ID: 000194900234
    FAST State              : Degraded
    Degraded Reason         : No DRVs
                              FAST VP is not licensed
    Error Message           : N/A
    Current Activities      : Idle
```

FAST state output values

Table 14 FAST State values

Value	Description
`Enabled`	FAST controller is enabled
`Disabled`	FAST controller is disabled
`Disabling`	The FAST controller is in transition from Enabled to Disabled, or Disable is in progress
`DisabledwithError`	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. Multiple degraded reasons can occur simultaneously. The Degraded Reason field may have a valid reason or it may display N/A.

Degraded reason values

- 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*.

Table 15 Current Activities values

Current Activities values	
Idle	FAST controller is currently idle.
RunningPlan	FAST controller is running a plan.
FetchingStats	FAST controller is fetching statistics
AnalyzingStats	FAST controller is analyzing statistics
PendingPlan	FAST controller has a scheduled plan and is waiting to start.

Time windows management

The `symtw` command manages time windows This command converts any existing time windows to the new time window format.

Refer to Managing Time Windows (Enginuity 5876 or lower) on page 605 for information about managing time windows.

Set FAST control parameters

Syntax

Set FAST control parameters, using the following syntax:

```
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>]
```

Examples

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 FAST recommendation is **100** hours, enter:

```
symfast -sid 207 set -control_parms -approval_mode AUTO_APPROVE -
min_perf_period 100 -workload_period 200
```

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.

FAST control parameters

Table 16 Fast control parameters for symfast CLI

Control parameter	Values — * indicates default value	Description
-approval_mode	**Auto_Approve*** **User_Approve**	Sets the mode of the FAST controller to automatic or user approval mode. If the FAST controller is set to **User_Approve** mode, it generates plans, but does not perform any movements unless the plans are approved by the user. The **-approval_mode** option sets the mode of the FAST controller and Optimizer. This parameter is for standard devices only.

Table 16 Fast control parameters for symfast CLI (continued)

Control parameter	Values — * indicates default value	Description
		Note The **-approval_mode** option replaces the **-mode** option. The -mode option will continue to work, although it will be deprecated. The **-mode** option is also used to set the CLI compatibility mode.
-min_perf_period	Minimum value — 2 (hours) Maximum value — current value of the workload period parameters (hours).	Specifies the amount of samples required before a FAST recommendation is made. Sampling time should be long enough (usually a week) for FAST to establish an adequate characterization of the typical workloads. A typical use for this parameter is to shorten the sampling time originally set.
-vp_data_move_mode	**Auto** **None**	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	Minimum value — 2 (hours) Maximum value — 672 (4 weeks)	Specifies the amount of workload sampling that FAST maintains for sample analysis. This is specified in units of time (hours). For example, this option can specify that the FAST controller maintain two weeks worth of workload samples for analysis.
-max_simult_devs	Minimum value — 2 Maximum value — 32	Determines the maximum number of devices that can be moved simultaneously. This parameter is for standard devices only.
-max_devs	Minimum value — 2 Maximum value — 200*	Determines the maximum number of devices that can be moved in a 24-hour period. This parameter is for standard devices only.
-vp_reloc_rate	Minimum value — 1 Maximum value — 10 Default value — 5	Indicates how aggressively FAST determines data movement for thin devices. This is similar to the max_simult_devs setting for standard devices. The lower the value of the VP relocation rate, the more aggressive FAST is in moving data. This setting does not affect the speed of the data movement; that is controlled by the VLUN QoS setting.
-swap_notvisible_devs	ENABLE DISABLE*	Indicates if the FAST controller can use host invisible devices (unmasked and unmapped) to

Table 16 Fast control parameters for symfast CLI (continued)

Control parameter	Values — * indicates default value	Description
		do a full swap, with devices in storage groups under FAST control, to improve performance of the storage group. This parameter is for standard devices only.
`-allow_only_swap`	ENABLE DISABLE*	Indicates if the FAST controller can only perform a full swap of devices and cannot move devices to unconfigured space. This parameter is for standard devices only.
`-pool_resv_cap`	Minimum value — 1 (%) Maximum value — 80 Default value — 10	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 pool (as a percentage of pool-enabled capacity) falls below the PRC, the FAST controller does not move any more data chunks into that pool. To move any new data chunks to the pool, the FAST controller must first move some data chunks from that pool to another pool to free up space. Enforcement of the PRC is best-effort; FAST may move data chunks to a virtual pool which results 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. **Note** To set the PRC of an individual pool, use the `set pool` command, as explained in Set pool attributes (Enginuity 5773 through 5876) on page 339. If the PRC is not set at the pool-level, the value set with `symfast` is the default.
`-vp_allocation_by_fp`	ENABLE DISABLE*	When set to ENABLE, for all thin devices managed by FAST, Enginuity chooses 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** This option is only supported by Enginuity 5876. All Enginuity versions prior to 5876 display `N/A` for this option.
`-time_to_compress`	Minimum value — 40 (days) Maximum value — 400 Default value — never (disabled)	Indicates the idle time FAST VP waits before it compresses devices automatically.

Table 16 Fast control parameters for symfast CLI (continued)

Control parameter	Values — * indicates default value	Description
		Note For businesses working on a quarterly basis, EMC recommends setting this value to 100 days to account for end-of-quarter activities.
`-fast_compression_rate`	Most aggressive — 1 Least aggressive —10 Default value — 5	Indicates how aggressively FAST looks for devices for compression.
`-compliance_mode`	ENABLE DISABLE*	Sets a higher priority for performing moves required to achieve compliance with the specified FAST policy percentages. This is a global FAST setting.

Display FAST control parameters

Syntax

To display the FAST control parameters, use the following syntax:

```
symfast [-sid SymmID] [-offline] list -control_parms
```

Note

If using the `-offline` option, populate the SYMAPI database with the storage group, tier, and FAST data using the `symcfg sync -fast` command:

Examples

To display the control parameters for VMAX array `432`, enter:

```
symfast -sid 432 list -control_parms
```

Sample output

```
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*
```

Note

The features requiring Enginuity 5876 Q42012 SR are marked with *.

Values for DP data movement mode are:

- `Auto_Approve` — Indicates the FAST controller is in automatic mode.
- `User_Approve`— Indicates the FAST controller is in user approval mode.

Values for VP data movement mode are

- `Auto` — Indicates the FAST controller is in automatic mode.
- `None` — Indicates the FAST controller will not perform any data moves on thin devices.

Display FAST plans

Description

FAST generates optimization plans approximately every hour, and these can be viewed at any time. Plans are subdivided into groups that execute serially. and each plan has a unique ID. 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)

Note

FAST VP does not display plans for thin devices. However, the Optimizer logs list data movement details (`symoptmz -sid` *SymmID* `read -log_type RUNTIME`).

Syntax

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 display using the **`symfast list -plan`** command. The **`Group Attributes`** field in the output lists the origin of the plan.

Examples

To list the FAST plan for VMAX array **`234`**, enter:

```
symfast list -sid 234 -plan
```

Sample output

```
Symmetrix ID                      : 000194900234

Plan ID                           : 12222011:154359
Plan Type                         : Auto Generated
```

```
Plan State                      : InProgress
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
    }
  }
```

For FAST DP plans, the plan listing also displays:

- source tier name
- source disk group number
- source RAID protection type
- Time started for each group
- Time completed for each group

FAST plan output values

Table 17 Group Attribute values

Value	Description
`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.

Table 18 Group State values

Value	Description
`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.

Table 19 Plan State values

Value (for USER_APPROVE mode, set to N/A for AUTO_APPROVE mode)	Description
`NotApproved`	Indicates this is a proposed plan for the FAST controller and needs user approval before the FAST controller can implement the plan.
`ApprovedWithSpecifiedTime`	Indicates the plan is approved and scheduled for a specific time.
`ApprovedWithDelay`	Indicates the plan is approved and is waiting in a queue.
`ApprovedWithConfigParameters`	Indicates the plan is approved and is scheduled according to 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 occurs when the user attempts to decline a running plan.

Approve a FAST plan

Description
When FAST is in *USER_APPROVE* mode, it generates plans approximately every hour, but does not act on them unless approved. A FAST plan must be approved as a whole and cannot be partially approved.

Note

The `approve` command is only used in *USER_APPROVE* mode.

Syntax
To approve a FAST plan, use the `symfast -plan approve -id PlanID` command . To identify the *PlanID*, use the `Plan ID` returned in the `symfast list -plan` output.

```
symfast -plan -sid SymmID [-i Interval] [-c Count] [-noprompt]
approve -id PlanID [-begin_at=TimeVal
```

Options

-begin_at=
Starts plan at a specified time. *TimeVal* format is **MMDDYYYY:HHMMSS**. If this option is not specified, FAST defaults to the time window definition

-i (interval) and -c (count)
Use to monitor FAST progress over specified period of time.

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
```

Decline a FAST plan

Description
A plan can be declined before or after approval. An approved plan can be declined while in progress, even in `AUTO_APPROVE` mode. An aborted plan deletes all groups except for the in-progress groups, which go to completion. 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`. The decline command is valid for both `AUTO_APPROVE` and `USER_APPROVE` mode.

Syntax
To decline a FAST plan, use the following syntax:

```
symfast -plan -sid SymmID [-i Interval] [-c Count] [-noprompt]
decline -id PlanID
```

FAST information display

The following FAST information displays and reports are available:

- Activity history
- Audit logs
- Compliance report

- Technology demand reports

Display FAST activity history

Syntax

To display the activity history, or the FAST plan for a VMAX array, use the following syntax:

```
symfast [-sid SymmID] list -history [-v]
[-start_date TimeVal] [-end_date TimeVal
```

Options

-start_date, -end_date

Use these options to filter history time range. *TimeVal* format is `MMDDYYYY:HHMMSS`.

Examples

To view the FAST activity history for VMAX array 234, enter:

```
symfast -sid 234 list -history
```

Sample output

```
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
    }
  }
```

FAST actions listed in audit log

The following FAST user actions 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 216

Technology demand report — Provides information about contention for tier resources. See "Technology demand reports" on page 218

Generate FAST compliance report

Description

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.

Options:

-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, otherwise capacities report in GB (default).

Syntax

To generate a report for the demand of a storage group and its associated FAST policy, use the following syntax:

Note

Prior to using the `-offline` option, the SYMAPI database must be populated with tier, storage group, and FAST data using the `symcfg sync -fast` command .

```
symfast -sid SymmID [-offline] list -association [-demand [-sg SgName
| -fp_name PolicyName][-mb]]
```

Examples

To generate a demand report, enter:

```
symfast list -sid 432 -association -demand
```

Sample output

```
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
```

FAST Compliance report output descriptions

Table 20 Compliance Report output values

Field Name	Description
OutofPolicy	Shows the devices in the storage group that currently do not reside on the tiers defined in the FAST policy.
Max SG Percent	Shows the limit (%) of the storage group per tier as defined in the FAST policy.
Max SG Demand	Shows the calculated upper limit (GB) for the storage group on the tier.
FAST SG Usage	Shows the current occupancy (GB) of the storage group in a tier.

Table 20 Compliance Report output values (continued)

Field Name	Description
	Note For storage groups associated with policies containing DP tiers, the limit for `FAST SG Usage` is based on the logical capacity of standard devices in the storage group only. Thin device capacity is not counted. For storage groups associated with policies containing VP tiers, the limit for `FAST SG Usage` is based on the allocaged capacity of thin devices in the storage group only. The capacity of any standard devices in the storage group is not counted, including any unallocated logical capacity of the thin devices.
`Growth`	Shows, as per the FAST policy, how much more the storage groupcan 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. **Note** For storage groups associated with policies containing DP tiers, the limit for `Growth` is based on the logical capacity of standard devices in the storage group only. Thin device capacity is not counted. For storage groups associated with policies containing VP tiers, the limit for `Growth` is based on the allocaged capacity of thin devices in the storage group only. The capacity of any standard devices in the storage group is not counted, including any unallocated logical capacity of the thin devices.
`RDF coordination`	Displays as N/A for FAST disk group associations, For FAST VP, displays as `Enable` or `Disable`.

Generate FAST technology demand reports

Description

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 are made.

Syntax

To generate a report for the demand (or details about) a technology type and its tiers, use the following syntax:

Note

Prior to using the `-offline` option, the SYMAPI database must be populated with tier, storage group, and FAST data using the `symcfg sync -fast` command

```
symfast [-sid SymmID] [-offline]list <-tech <EFD | FC | SATA | ALL> -
external>
-demand [-v] [-dp | -vp][-allocated]
```

Options

-allocated
By default, the maximum storage group demand is calculated using the logical (configured) capacity of thin devices under FAST control. If `-allocated` option is specified, To calculate the Max SG Demand using the allocated capacity of thin devices under FAST control, use the `-allocated` option .

-external
Use for reporting external tiers. When using this option do not use the `-technology` (`-tech`) option.

-technology
Specifies the technology type for the tier.

-dp, -vp
Specifies tier type. `-dp` is the default. If this option is not specified, and the array only contains VP tiers, then VP tiers are reported.

Examples

To generate a demand report for VP tiers on SATA, enter:

```
symfast list -sid 432 -tech SATA -demand -vp
```

To generate a demand report for VP external tiers, enter:

```
symfast -sid 432 list -external -demand -vp
```

The VP demand report displays information only for VP tiers and for storage groups associated with FAST policies containing VP tiers.

Sample output

For VP tiers SATA:

```
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
```

For external tier reports, the `Technology` displays as N/A.

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.

FAST Technology demand report output descriptions

Table 21 Technology Demand Report output values

Field Name	Description
Tier	Names of VP tiers.
ATTR	Status of the VP tier. Tiers can have one of 3 possible attributes: • (F) — In a FAST policy associated with a storage group. • (P) — In a FAST policy or policies where none of the FAST polices are associated with a storage group. • (N) — Not in any FAST policy.
Target Prot	Target protection for the tier.
Tier Enabled	Total logical capacity enabled for the tier.
Tier Free	The enabled logical capacity minus the used logical capacity, minimum is 0.

Table 21 Technology Demand Report output values (continued)

Field Name	Description
`Tier Used`	Total pool allocated capacity for all thin pools in the tier. Allocated capacity on all DATA devices are counted, including DATA devices not enabled; therefore `Tier Used` may be greater than `Tier Enabled`.
`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 not included in any VP Tier.
`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 is 0.
`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

Manage Configuration Changes

This chapter describes configuration change concepts and explains how to perform change operations using the `symconfigure` command.

- Configuration change management overview 224
- Verify and check sessions 226
- Abort configuration session 227
- Configuration change guidelines 227
- Supported configuration operations 230
- Set array attributes 234
- Set SLO name (HYPERMAX OS 5977 or higher) 235
- External disk group management 236
- Create devices (HYPERMAX OS 5977 or higher) 236
- Create devices (Enginuity 5773 through 5876) 240
- Copy devices 249
- Metadevice management (Enginuity 5773 - 5876) 251
- Configure thin metadevices (Enginuity 5773 - 5876) 259
- Convert devices 260
- Set device attributes 265
- Set device geometry on Solaris hosts (Enginuity 5773 - 5876) 267
- Set device identifiers 270
- Delete devices (HYPERMAX OS 5977)) 271
- Delete devices (Enginuity 5773 - 5876) 272
- Device reservation management 273
- Device pool management (HYPERMAX OS 5977) 276
- Device pool management (Enginuity 5773 - 5876) 276
- Map devices 281
- Map CKD devices (Enginuity 5773 - 5876) 283
- Unmap devices 286
- Introduce devices to a host 287
- Configure dynamic SRDF-capable devices (Enginuity 5773) 289
- Set SRDF group attributes 291
- Swap RA groups 291
- Convert directors (Enginuity 5876 only) 291
- Port to director emulation support (HYPERMAX OS 5977) 292
- Set port characteristics 296

Configuration change management overview

Use the SYMCLI `symconfigure` command or the `symdev` command to make configuration changes to a locally-connected array or to an RDF-linked array. This command is run from the local host and performs control operations on arrays, as well as array devices, groups, directors, and ports.

Note

The `symdev create`, `symdev modify`, and `symdev delete` commands are only supported on arrays running HYPERMAX OS 5977.

Array controls include:

- Setting array-wide metrics
- Determining what type of devices the array will support, such as RAID 6 devices

Device controls include:

- Creating, modifying, and deleting devices
- Mapping and masking devices
- Configuring device pools
- Reserving devices
- Releasing device reservations

Configuration change session rules

The following rules apply to change sessions and concurrent change sessions:

- Concurrent Provisioning — Up to four concurrent configuration change sessions can run at the same time, when they are non-conflicting. 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

Description

Configuration changes that are submitted to the array are processed in a session. The `symconfigure` command has several formats for executing array configuration changes. The command file format can contain various command entries terminated with a semicolon (;). Multiple changes can be specified 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.

Refer to the EMC Solutions Enabler SYMCLI Command Reference for the complete manpage description of the `symconfigure` command.

Options

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.

Examples

There are various ways to execute the `symconfigure` command.

- 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
```

Verify and check sessions

Description

Use the `symconfigure verify, preview,` and `prepare` arguments to verify and check a change session that is acting on a specified `symconfigure` command file. The `commit` argument performs these same checks, then attempts to execute the specified configuration change.

The `symconfigure` command modifies configuration operations and components. A lock may be taken out by the specified VMAX array configuration server while the configuration change session is active. Use the `query` option to monitor the stages of session processing. Not all stages are always executed. Use caution when controlling which stages are to be completed, to allow checking and debugging of the command file before the changes are implemented.

Options

verify
: Determines if a VMAX array can be modified, as there are restrictions as to what state the configuration must be in before allowing changes to be applied.

preview
: Verifies that each individual change is valid and the syntax is correct, and then terminates the session without change execution.

prepare
: Performs the preview checks, validates the change operation (devices are in correct state, etc.), then terminates the session without attempting to make the configuration change. For arrays running Enginuity 5773 through 5876, `prepare` is not supported for SAVE device pool sessions.

commit
: Completes all checks and verifications, and then attempts to make the requested configuration changes in the specified array.

query
Checks the status of any configuration change session or whether there is a configuration session running. This 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 modifying a configuration by rearranging the placement of data.

Example
To check the status of the change session for the next two minutes, every 10 seconds on array **345**, enter:

```
symconfigure -sid 345 query -i 10 -c 12
```

Abort configuration session

Options

-session_id
Aborts specified session when multiple sessions are running on the array. If -`session_id` is not specified the abort command displays each running session and prompts for the session ID.

Examples
The `abort` option allows you to stop a configuration session. To abort a change session on a specific array, enter:

```
symconfigure -sid 12345 abort
```

To abort change session 100, enter:

```
symconfigure -sid 343 abort -session_id 100
```

Restrictions

- 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 or gone beyond this point.

Configuration change guidelines

Understand your array configuration before making configuration changes. Any changes can impact stored data.

Note

If you have problems with a new configuration, contact the EMC Customer Support Center for assistance in reverting to your previous configuration.

Ensure that all critical data is preserved and safe when creating new or changing device configurations, and do not store data on any device that is not mirrored, or RAID-protected. All configuration changes 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.

Verify viable array configuration

Syntax

Verify that the current array configuration is a viable configuration for host-initiated configuration changes. To verify the current array configuration is ready for changes, use the following syntax:

```
symconfigure verify -sid SymmID
```

Check for free physical disk space

Syntax

Before creating new devices, check for free physical disk space using the following syntax:

```
symconfigure list -freespace [-units CYL | MB] -sid SymmID
```

Note

Free disk space on unformatted disks is shown as available for all emulation modes. New devices are 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 syntax formats:

```
symdev list -sid SymmID -da all -space
```

```
symdisk list -sid SymmID
```

Stop I/O activity on affected devices

Description

Configuration changes begin only after issuing the `commit` action. Some classes of change operations may or may not impact current I/O. When possible, before issuing the `commit` action, stop I/O activity on the affected devices.

Note

If I/O activity on an affected device occurs before or during a `commit` action, the action may fail. At the very least, heavy I/O activity on unaffected devices impacts how long it takes to commit changes.

Syntax

If required, use the following syntax to set the impacted devices for change to be *Not Ready*:

```
symdg -g DgName not_ready [LdevName [LdevName...   ]]
```

Mixed array environments

Before issuing a `symconfigure` command to an array containing both open system and mainframe devices, verify that the mainframe Missing Interrupt Handler (MIH) period for each mainframe server attached to the array is set to at least two 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, consult the mainframe system administrator.

Blocked operations during active change session (Enginuity 5876 or lower)

The following `symconfigure` operations are 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

Update host with device mapping information

After a configuration change, the host's device information must be updated. Attempting host activity with a device after it has been removed or altered, but before updating the host's device information, can cause host errors. To update the host:

Procedure

1. Commit a `symconfigure map` operation.
2. Run the utilities specified for the host platform as described in Introduce devices to a host on page 287.
3. Issue the `symcfg discover` command to update the SYMAPI database with the new device mapping information.
4. Resume I/O activity.

Supported configuration operations

Table 22 Configuration change operations

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	✓	✓	✓	✓	✓
Set device attributes:					
VCMdb	✓				
ACLX		✓	✓	✓	✓
RDB_Cksum	✓				
dyn_rdf	✓	✓	✓	✓	
dyn_rdf1_only	✓	✓	✓	✓	
dyn_rdf2_only	✓	✓	✓	✓	
SCSI3_persist_reserv	✓	✓	✓	✓	✓
RCVRPNT_TAG			✓	✓	

Table 22 Configuration change operations (continued)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
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			✓	✓	
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		✓	✓	✓	

Table 22 Configuration change operations (continued)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
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	✓	✓	✓	✓	
Set front-end port attributes	✓	✓	✓	✓	✓
Associate/disassociate a port to director emulation					✓
Set array-wide parameters:					
auto_meta	✓	✓	✓	✓	
auto_meta_config	✓	✓	✓	✓	

Table 22 Configuration change operations (continued)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
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				✓	✓

Set array attributes

Syntax

Use the following command to set certain attributes to control array behavior:

```
set Symmetrix MetricName=MetricValue          [,
MetricName=MetricValue];
```

Options

MetricName

The metric to be set. Possible metric names are listed in Table 23 on page 234 and Table 24 on page 234.

Table 23 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 24 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 30 on page 252.
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 30 on page 252.
pav_alias_limit	If PAV is enabled, specifies the maximum number of aliases that can be assigned to a device. Possible values are:

Table 24 Possible metric names (Enginuity 5773 through 5876) (continued)

	Enginuity 5771/5772: 1-255 Enginuity 5773 to 5876: 1-127
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.

View the array metrics

Syntax

To view the current settings for array metrics, use the following syntax:

```
symcfg -sid SymmID -v list
```

Set SLO name (HYPERMAX OS 5977 or higher)

Syntax

To set to a user-defined SLO name or reset to the default name, use the following syntax with `symconfigure`:

```
set slo <SLOName=NewSLOName|BASE NAME>
```

Examples

To set the SLO default name to a user-defined name, enter:

```
symconfigure -sid 234 commit -cmd "set slo Platinum
name=Emc_Platinum;"

Execute a symconfigure operation for symmetrix '000000001234' (y/
[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change
    session..............Established.
    Processing symmetrix 000000001234
    {
          set slo Platinum name=Emc_Platinum;
    }
    Performing Access checks ..............................Allowed.
      . . .
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

To reset the SLO default name to the default name, enter:

```
symconfigure -sid 234 commit -cmd "set slo Emc_Gold name=BASE NAME;"
```

Rules and restrictions for set/reset SLO name

The following security privileges are required to execute this command:

- Required Access type: CFGSYM
- Required Authorization Rights: Storage Admin

The following naming rules apply:

- Names cannot exceed 32 characters
- Alphanumeric characters only
- Hyphens (-) and underscores (_)are allowed, but not as the leading character

This command will fail if:

- The `slo_name` value is a NULL value, or if the specified SLO name does not exist.
- The new SLO name supplied is already in use, as either a base SLO name or a SLO current name.

  Note

  Names must be unique when folded to upper case.

- A reset to a base SLO is performed and a user-defined SLO does not exist.

External disk group management

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 Federated Tiered Storage on page 361 for eDisk configuration details.

Create devices (HYPERMAX OS 5977 or higher)

Description

Use the `symconfigure create dev` command or the `symdev create` command to create new devices.

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

Syntax

To create one or more devices, use the following `symconfigure` syntax:

```
symconfigure 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_CK[,...]]
                    [, device_name='<DeviceName>'[,number=<n |
SYMDEV> ]];
```

To create one or more devices, use the following `symdev` syntax:

```
symdev -sid <SymmID> [-noprompt]
       create -tdev -cap <#> [-captype <cyl|mb|gb|tb>] [-N <#>] [-bcv]
    create <-gk|as400_gk|-pedev> [-N <#>]
```

Options

count/#
Indicates the number of devices to create.

size/cap
Specifies the size of the device needed in number of cylinders (default), megabytes, gigabytes, or terabytes.

Table 25 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.

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.

Examples

```
symdev create -sid 005 -cap 10000 -nop –tdev

Create operation succeeded for devices: 01ff0.
```

Restrictions when setting device names (HYPERMAX OS 5977)

For arrays running HYPERMAX OS 5977, 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 (HYPERMAX OS 5977)

For arrays running HYPERMAX OS 5977, the following restrictions apply for creating and managing VMAX devices and storage pools:

- Arrays running HYPERMAX OS 5977 allow the creation of 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.
- SRDF device types cannot be created or converted 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. For HYPERMAX OS 5977 and higher, gatekeeper devices can be added to storage groups using the syntax `sg=SgName`.

All information about gatekeepers can be found in the *Solutions Enabler Installation Guide* and in 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.

Modify device capacity (HYPERMAX OS 5977 or higher)

Syntax

To modify device capacity, use the following syntax:

```
symdev -sid <SymmID> [-noprompt]
       modify -tdev -cap <#> [-captype <cyl|mb|gb|tb>]
          -devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
             [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>
```

Examples

```
symdev modify 1fe0 -sid 005 -cap 20000 -nop -tdev

Modify operation succeeded for devices: 1fe0
```

Create devices (Enginuity 5773 through 5876)

The creation of certain device (configuration) types using the `symconfigure` command may require two sessions. Table 26 on page 240 describes the sessions required to create a device type and/or convert the various types of devices from one configuration type to another.

For information on creating dynamic-RDF capable devices, refer to Configure dynamic SRDF-capable devices (Enginuity 5773) on page 289.

When an SRDF device is created 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

Create SAVE devices and DATA devices using the same procedure used for standard protected devices, with the additional specification of `attribute=savedev` or `attribute=datadev`.

Multi-session device creation (Enginuity 5773 through 5876)

Table 26 Steps to create a device

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	
2-Way-DRV-Mir	2-Way-DRV-Mir	

Table 26 Steps to create a device (continued)

Desired device type	Session 1 create	Session 2
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[dd e]+R5	RDF2+R5	Add R1 mirror
RDF21[de]+R6	RDF1+R6	Add R2 mirror
RDF21[de]+R6	RDF2+R6	Add R1 mirror
RDF21[de]-Mir	RDF2-Mir	Add R1 mirror
RDF22[e f]+R5	RDF2+R5	Add R2 mirror
RDF22[ef]+R6	RDF2+R6	Add R1 mirror
RDF1+TDEV	RDF1+TDEV	
RDF2+TDEV	RDF2+TDEV	
RDF1-BCV+TDEV	RDF1+TDEV	Convert to RDF1-BCV+TDEV
RDF2-BCV+TDEV	RDF2+TDEV	RDF2-BCV+TDEV
DLDEV[f]		

Table 26 Steps to create a device (continued)

Desired device type	Session 1 create	Session 2
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 or higher.
c. Not supported on Enginuity 5771 or higher.
d. Supported only on Enginuity 5773 or higher.
e. To create an R21 or R22 device, use `add rdf mirror` with the `symconfigure` command. This command fails if the device is a Data Domain device.
f. Supported only on Enginuity 5874 or higher.

Create one or more devices (Enginuity 5773 - 5876)

Syntax

To create one or more devices, use the following syntax:

```
symconfigure 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=
   [starting] base_address= <cuu_address>]...]
[[device_attr =
  [SCSI3_persist_reserv | ACLX |DIF1 |AS400_GK]]...]
[,device_name=DeviceName  [,number=<n | SYMDEV>]];
```

Options

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 27 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 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 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 26 on page 240 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 26 on page 240.

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.

Table 28 Number of data members for RAID 5 and RAID 6 devices

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 26 on page 240.

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.

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:

Table 29 Local and remote device assignments

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 Create devices (Enginuity 5773 through 5876) on page 240.

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 Set device attributes on page 265 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 (Enginuity 5773 - 5876)

For Enginuity 5773 through 5876, 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 (Enginuity 5773 - 5876)

- Unprotected 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.
- Devices can be created with a variety of emulation types: Fixed Block Architecture (FBA), Celerra FBA or VME512 FBA, and CKD for z/OS environments.

- Creating or converting to SRDF device types is not allowed 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.
- Partial allocation of thin devices is not supported.
- 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.

AS400 devices

- 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 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

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, so 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 configurations

- 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 configurations

- SAVE devices work in conjunction with virtual devices and can be unprotected, mirrored, or RAID 5/RAID6.
- 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` when creating a new device.

Copy devices

Copying the attributes of an existing device into available disk space configures new devices based on the copied attributes. To configure new devices this way, 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, copied attributes can be changed. For example, copied attributes of a standard device, can be changed to BCVs on the new device.

Copy a similar device

Syntax

To configure a device by copying a similar device, use the following syntax:

```
symconfigure 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.

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 in Create one or more devices on page 242.

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 in Create one or more devices on page 242.

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 remote 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`*

Metadevice management (Enginuity 5773 - 5876)

Note

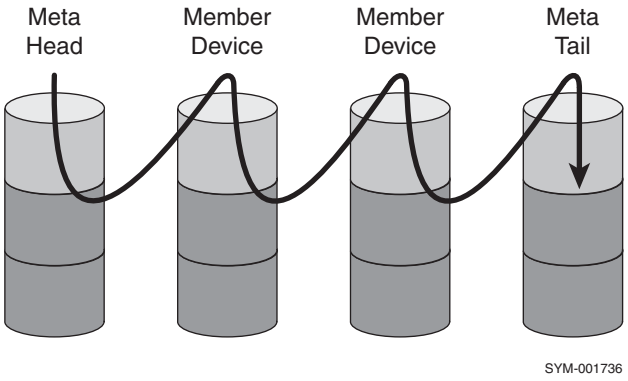
HYPERMAX OS 5977 does not support metadevices.

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 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.

Figure 7 Concatenated metadevices



Metadevice size requirements (Enginuity 5773 - 5876)

Metadevices can contain as many as 255 members. The maximum size tested and supported by EMC is 60 terabytes. Metadevices can consist of nonsequential and nonadjacent volumes.

Table 30 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
Enginuity 5874 and 5876	262668	240	262669	262668
Enginuity 5773	65520	59	65521	65520

Meta performance (Enginuity 5773 - 5876)

To enhance disk system functionality and improve throughput and performance, the array supports:

- Grouping individual physical disk devices into a metadevice
- Metadevice addressing
- Multiple I/O drive queues for metavolumes

Metadevice data addressing (Enginuity 5773 - 5876)

The array supports two types of metadevice data addressing:

- Concatenated devices
- Striped devices

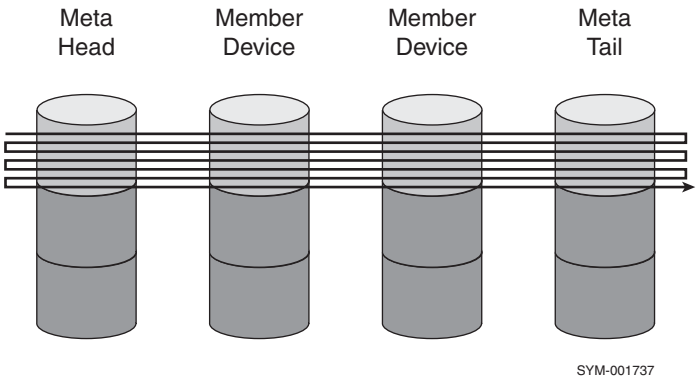
Concatenated devices

Concatenated devices are volume sets, organized with the first byte of data at the beginning of the first device. 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

Striped metadevice addressing 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. 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 8 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 balances the I/O activity between the disk devices and the array disk directors.

If a stripe size is not specified when creating a striped meta, then the stripe size is the default of 1920 blocks of 512 bytes each:

The sum of the metamember's individual cylinder size and the metahead cylinder size must not match. Use the `symdev show` command to determine the usable cylinder size.

Metadevice creation methods (Enginuity 5773 - 5876)

Metadevices can be created in either of the following ways:

- Automatically, with the create device command, as explained in Create devices (Enginuity 5773 through 5876) on page 240.

  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 Form metadevices (Enginuity 5773 - 5876) on page 253.

Form metadevices (Enginuity 5773 - 5876)

If the automatic meta creation feature is not enabled, use the `form meta` command to form metadevices. Metadevice configurations require two sessions (form and convert) as shown in Table 31 on page 253.

Table 31 Steps to create an SRDF BCV metadevice

Desired device type	Session 1 `form`	Session 2 `convert`
Meta RDF1-BCV	Meta-BCV	-->Meta RDF1-BCV

Table 31 Steps to create an SRDF BCV metadevice (continued)

Desired device type	Session 1 `form`	Session 2 `convert`
Meta RDF2-BCV	Meta-BCV	-->Meta RDF2-BCV

Note

Cacheless Read Miss support can return an error if the devices specified to form a meta contain different *Cacheless Read Miss* modes.

When initially creating a metadevice, 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 automatically selects members from the pool of unmapped devices that match the metahead in size, emulation, and configuration type.

Note

Confirm that devices being considered for a metadevice have the SCSI-3 persistent reserve flag disabled.

Form a metadevice (Enginuity 5773 - 5876)

Syntax

To form a metadevice, use the following syntax:

```
symconfigure form meta from dev SymDevName,config=MetaOption
[, stripe_size=<MetaStripeSize>[cyl]][, count=<member_count>];
```

Options

config
Specifies the meta configuration type. Possible values are `concatenated` or `striped`.

stripe size
Specifies the size of the striped metadevice. This value is 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 a stripe size is not specified when creating a striped meta, all Enginuity codes consider the default stripe size as 1920 blocks of 512 bytes each. For information on metavolume stripe sizes, refer to Striped devices on page 253.

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. If using the `add dev` command to specify members, omit this option.

Create RDF metadevices (Enginuity 5773 - 5876)

Description

Use the `create dev` command to create static and dynamic SRDF metadevices.

Options

auto_meta
To use this option, both the local and the remote arrays must be running Enginuity 5773 through 5876 and the local array must have `auto_meta` enabled. Any `auto_meta`settings on the remote array are ignored.

min_auto_meta_size
Specifies the minimum size of the SRDF metadevice.

Example
To create a two-member metadevice on both a local array and remote array. enter:

Note

Assume the local array is enabled for `auto_meta` and `min_auto_meta_size` is set to 30000.

```
symconfigure create dev count=1, size = 60000, emulation=FBA,
config=RDF1,ra_group=20,remote_config=RDF2,dynamic_capability=dyn_rdf;
```

Restrictions
SRDF devices cannot be used to form metadevices in one step. The SRDF devices must first be converted to non-RDF devices (as shown in Table 31 on page 253), 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 is established, it cannot be modified in one step. If any modifications are required, it will have to be converted again to a non SRDF device first.

Create AS/400 metadevices (Enginuity 5773 - 5876)

Description
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.

Options

auto_meta
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 attempts to create an AS/400 non metadevice, and rejects a request to create an AS/400 metadevice.

Example
To create an AS/400 metadevice on an array with emulation AS/400_M4328_50, enter:

```
create dev count=1, emulation= AS/400_M4328_50, config=2-Way-Mir;
```

This creates an 8-member metadevice where total cylinder size is 143576 (native mode).

Meta configuration restrictions (Enginuity 5773 - 5876)

A non meta device being converted concatenated metadevice can be mapped and online to the host during the operation.

Note

A non metadevice being converted to a *striped* metadevice must be unmapped before it can be converted.

The following restrictions apply 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. Inner members cannot be removed.
- 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 (SRDF devices need to be converted 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.
- 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 31 on page 253 for converting a metadevice for SRDF operation). The configuration manager cannot form a metadevice from SRDF devices that are RDFA-protected (that is, these devices cannot be converted to non-RDF, form the metadevice, and convert them back to RDF).

Add members to a concatenated metadevice (Enginuity 5773 - 5876)

Syntax

To add additional members to an existing concatenated metadevice, use the following syntax:

```
symconfigure add dev SymDevName[:SymDevName] to meta SymDevName;
```

Example

To add devices 00013 and 00014 to concatenated meta 00010, enter:

```
symconfigure 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 devices added to a meta contain different Cacheless Read Miss modes.

Add members to a striped metadevice (Enginuity 5773 - 5876)

Description

Solutions Enabler supports expanding a striped thin metadevice with data protection.

Syntax

To add additional members to an existing striped metadevice, use the following syntax:

```
symconfigure add dev SymDevName[:SymDevName] to meta SymDevName[,
protect_data=[TRUE | FALSE],
 bcv_meta_head=SymDevName];
```

Options

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`, 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. Instead, protect data by using a mirrored BCV device.

bcv_meta_head

When adding new members to an existing striped metadevice, to protect data on the metadevice, 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.

Example

To add devices **00013** and **00014** to striped meta **00010**, enter:

```
symconfigure add dev 00013:00014 to meta 00010, protect_data=TRUE,
bcv_meta_head=00CA;
```

Remove metamembers (Enginuity 5773 - 5876)

Syntax

To remove a metamember or members from the tail of a concatenated metadevice, use the following syntax:

```
symconfigure remove dev SymDevName[:SymDevName] from meta SymDevName
```

Example

To remove device 00014 from meta 00010, enter:

```
symconfigure remove dev 00014 from meta 00010;
```

Dissolve a metadevice (Enginuity 5773 - 5876)

The `symconfigure` command file uses 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 of the metahead devices, with the device numbers, or use the device numbers of the meta heads within a device range. The command skips 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 is returned.

Syntax

To dissolve one or more metadevices, use the following syntax:

```
symconfigure dissolve meta dev SymDevName[:SymDevName]
[[,SymDevName[:SymdevName]]...];
```

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 using the file `dissolve_meta.cmd`,enter:

```
symconfigure -sid 266 commit -file dissolve_meta.cmd -v -noprompt
```

Note

A `dissolve` action removes the metahead and all its members.

Metadevice data preservation (Enginuity 5773 - 5876)

When managing metadevices, data is not written directly to the devices involved, so the original data is essentially preserved. However, because data is reorganized after device management actions finish, the host cannot access all of the original data. The attached host is responsible for handling the change to the physical geometry of the device.

Note

The expansion of striped metadevices is an exception to this rule. In this case, provide a BCV meta to preserve the original data, which is rewritten to the newly-configured striped metadevice.

The following table lists data management operations and their effects on data integrity.

Table 32 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]

Table 32 Metadevice data preservation (continued)

Metadevice type with data	Operation	Data integrity preserved status
Metadevices concatenated	Dissolving to non meta	Not preserved [b]
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 non-configurable option

NOTICE

Dissolving striped or concatenated devices causes data access loss. Adding to concatenated metadevices preserves the metadevice data, but makes the new members's original data inaccessible. Removing concatenated members causes data access loss on the removed member. Use these commands with great caution.

Configure thin metadevices (Enginuity 5773 - 5876)

Note

HYPERMAX OS 5977 does not support thin metadevices .

Use the `create dev, form meta`, `dissolve meta`, and `convert meta` commands to configure a thin metadevice. Follow these guidelines when creating thin metadevices.

- For a striped metadevice:
  - All members must be unmapped, unbound, and have the same device size.
- For a concatenated metadevice:
  - On arrays running Enginuity versions 5874 or lower,only unmapped thin devices can be 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.
- For a thin metadevice:
  - Only thin devices can be used to create a thin metadevice.
  - Mixing thin and non thin devices to create a meta is not allowed.

Note

Only unbound thin metadevices can be dissolved.

Thin BCV metadevice guidelines (Enginuity 5875 - 5876)

Solutions Enabler with Enginuity 5875.198.148 and 5876 supports using a thin BCV metavolume while expanding striped thin metavolumes.

The following guidelines apply:

- 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.
- Using 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 Create thin devices (Enginuity 5773 through 5876) on page 335).

  As a part of the expansion process, if the original metadevice was bound to a thin pool, the new metamember is bound to the same pool.

Bind a thin metadevice to a pool (Enginuity 5773 - 5876)

When binding a thin metadevice to a pool, use the `symconfigure` command to bind the metahead and give one preallocated size. Solutions Enabler distributes the preallocated size among all of the metamembers.

The distribution depends on the type of the metadevice, as follows:

Striped metadevice — The preallocated size is 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 three members, each of 100 GB, Solutions Enabler allocates 100 GB to the head and 75 GB to the 2nd member and 0 GB to the last member.

Refer to Configuring Virtual Provisioning on page 331 for more information about binding thin devices.

Convert devices

Syntax

To convert a device configuration type, use the following syntax:

```
symconfigure 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]];
```

Note

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.

Options

SymDevName
Specifies the 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 33 on page 263 lists allowable configuration conversions.
For arrays running Enginuity 5773 through 5876, possible configuration conversions are shown in the first column of Multi-session device creation (Enginuity 5773 through 5876) on page 240.

emulation
Indicates the device's emulation type.

ra_group
Specifies the RA group number in the SRDF environment.

remote_dev
Specifies the name of the remote array device targeted for change. If a range of *SymDevNames* is specified in the first line of the convert statement, the remote *SymDevName* value is increased incrementally to arrive at the corresponding device number.

invalidate_opt
Indicates the SRDF device to invalidate so 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 is assigned to any device created as a result of removing any mirror(s). If not provided, the original MVS SSID is assigned when available. If the MVS SSID group is full, supply a new MVS SSID. Only one `remote_mvs_ssid` can be used in a session; it is 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 is assigned to any device created as a result of removing any mirror(s). If not provided, the original MVS SSID is assigned when available. If the MVS SSID group is full, supply a new MVS SSID. Only one **`mvs_ssid`** can be used in a session; it is 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.

Example

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:

```
symconfigure convert dev 0001C:0001D to RDF1-BCV, ra group=1,
remote_dev=0001c, invalidate=R1, start_copy=YES;
```

Device conversion restrictions

Restrictions

The following restrictions apply when 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.
- One member of a raidset cannot be converted to unprotected without converting all 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. Changes are automatically applied to metamember(s).

Valid thin device conversions

Thin devices can be converted to other thin device configurations, as shown in the following table.

Note

Table 33 Thin device conversions

Original thin device	Converted to
TDEV [a]	RDF1+TDEV
TDEV [a]	RDF2+TDEV
TDEV	BCV+TDEV
BCV+TDEV [b]	TDEV
RDF1+TDEV [a]	TDEV
RDF2+TDEV [a]	TDEV
BCV+TDEV [a]	RDF1-BCV+TDEV
BCV+TDEV [a]	RDF2-BCV+TDEV
RDF1+TDEV [a]	RDF1-BCV+TDEV
RDF2+TDEV [a]	RDF2-BCV+TDEV
R1-BCV+TDEV [a], [c]	R1+TDEV
R2-BCV+TDEV [a], [c]	R2+TDEV
R1+TDEV [c]	R1-BCV+TDEV
R2+TDEV [c]	R2-BCV+TDEV

a. Not for VMAX 10K Series systems; all devices are dynamic RDF capable by default.
b. BCV+TDEV to TDEV is the only supported conversion for a BCV+TDEV device.
c. Requires Enginuity 5875 Q32001 SR

Convert a thin device

Example

To convert a thin device TDEV to an RDF1 thin device, enter:

```
symconfigure convert dev 00015 to RDF1+TDEV;
```

To create a thin R1 BCV device by converting a thin BCV device, enter:

```
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
```

Thin device recommendations to maximize I/O activity

Adhere the following 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 - 5876)

When converting devices that are currently part of a device group, some groups will be declared invalid after particular operations.

For example, if you are performing TimeFinder operations and have placed one standard (STD) device and one TimeFinder (BCV) device in a group. When `symconfigure` allows

the BCV device to change to a non-BCV device, the conversion leaves the group in an invalid state.

To correct this:

- Remove the TimeFinder (BCV) device from the group before starting 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.

Convert a metadevice (Enginuity 5773 - 5876)

Description

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.

Syntax

To convert a metadevice's configuration from concatenated to striped.

```
symconfigure convert meta SymDevName config=MetaOption
[, stripe_size=MetaStripeSize>[cyl]],
[, protect_data=[TRUE|FALSE],
[bcv_meta_head=SymDevName];
```

Options

strip_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 deveice name (or thin device name).

Examples

To convert a concatenated metadevice 00030 to a striped metadevice and preserve data during the conversion using BCV metadevice 01F, enter:

```
symconfigure 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.

Supported thin metadevice conversions (Enginuity 5773 - 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.

Set device attributes

Syntax

To set the device attributes or emulation of a number of devices in a range, use the following form:

```
symconfigure set dev SymDevName[:SymDevName]
      [emulation=EmulationType]
      [identity = NO identity]
      [attribute=[NO] device_attr];
```

Options

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.

Examples

To convert five devices (00015 to 00019) to Celerra FBA emulation, enter:

```
symconfigure set dev 00015:00019 emulation=CELERRA_FBA;
```

Device attribute values

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 attribute 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.

AS400 Gatekeeper attribute restrictions

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

Set device geometry on Solaris hosts (Enginuity 5773 - 5876)

To set the geometry for a disk on arrays managed by a Solaris host, use the `symconfigure` command, which is executed using a command file.

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, `Symm-6` represents arrays with Enginuity 56*xx* and earlier, and `Symm-7` represents arrays with Enginuity 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, 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`, changing the geometry on the replicated `Symm-7` device may be required. If the `Symm-7` device were to be cloned, the clone device does not automatically receive the emulated geometry—it must be explicitly set.
- Keep in mind the exact geometry of the source disk and ensure that before the migration or replication occurs, this geometry is set for the target disk.

FBA Geometry Emulation

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.

Set device geometry (Enginuity 5773 - 5876)

Syntax

Note

This feature is for Solaris hosts using the `format` utility.

The syntax for setting the device geometry is as follows:

```
symconfigure set dev SymDevName[:SymDevName]
   geometry [= SYMM-6 | = SYMM-7 | = CLARIION]
   cyls = n;
set dev SymDevName[:SymDevName]
   geometry = <NO GEOMETRY>;
```

Options

SYMM-6
The device geometry in arrays running Enginuity 56xx and earlier (64 sectors/track, 15 tracks/cylinder).

SYMM-7
The device geometry in arrays running Enginuity 57xx and higher (128 sectors/track, 15 tracks/cylinder).

CLARIION
The CLARiiON disk geometry (10 sectors/track, 16 tracks/cylinder; each sector is 512 bytes).

cyls=*n*
The device cylinder size.

NO GEOMETRY
Clears the current geometry if the device has an emulated geometry set.

Restrictions

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

Show device geometry

Syntax

Note

This feature is for Solaris hosts using the `format` utility.

Use the following commands to display disk geometry:

```
symdg list -v
symdev list -v
sympd list -v
symdev show -geometry
sympd show -geometry
```

Examples

```
symdev -sid 016 show 0317 -geometry
```

Sample output

```
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 `Effective Device Geometry` field can have four possible values:

Table 34 Values for Effective Device Geometry

Value	Description
`User Defined`	Indicates that the device has user-defined geometry.
`Native`	Indicates the current device geometry is the same as the native geometry for the array.
`Array wide emulation`	Indicates that the array-wide `FBA Geometry Emulation` flag 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.

Set device identifiers

Description

Solutions Enabler supports device identifier management on arrays. This support allows defining 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.

A device name or identifier can be set on a single device, a range of devices, or multiple ranges of devices. Device identifiers do not have to be unique for all devices.

Syntax

The format of the command file for setting device identifiers follows:

```
symconfigure set dev SymDevName[:SymDevName]
    [[device_name = 'DevName'] | [NO DevName]]
    [[hp_identifier = 'hp_id'] | [NO hp_identifier]]
    [[vms_identifier = vms_id] | [NO vms_identifier]];
```

Restrictions

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`.

Note

The `device_name` parameter is supported only for Enginuity 5874 or higher.

- Restrictions for `device_name`:
  - *DevName* 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.
- Restrictions for `hp_id`:
  - *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`:
  - A *vms_id* can be a number between 0 and 32766.

Identifier exclusions

Device identifiers cannot be set 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)

View device identifiers

Use the `symdev list` command to view device identifiers.

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.

Delete devices (HYPERMAX OS 5977))

Description

Use the `symconfigure delete dev` command or the `symdev delete` command to delete devices.

Syntax

To delete one or more devices, use the following syntax:

```
symconfigure delete dev SymDevName[:SymDevName]
```

```
symdev -sid <SymmID> [-noprompt]
          delete -devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
             [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>
```

Examples

To delete device `00015` from array `345`, create a command file containing the following:

```
delete dev 00015;
```

Then commit the option using the `symconfigure` command:

```
symconfigure -sid 345 -file delete_dev.cmd -v -noprompt commit
```

Using the `symdev` command:

```
symdev delete 1f00 -sid 005 -nop

Delete devices operation succeeded.
```

Restrictions (HYPERMAX OS 5977)

- The `delete` commands are blocked if the source device specified by *SymDevName* is a TDAT device.
- The device must not be mapped to a front-end port.

Delete devices (Enginuity 5773 - 5876)

Syntax

To delete one or more devices, use the following syntax:

```
symconfigure delete dev SymDevName[:SymDevName][,raidset = TRUE |
FALSE];
```

Note

The `raidset` option is available with arrays running Enginuity 5773 through 5876.

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.

Examples

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

- 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

Device reservation management

Use the configuration change functionality to reserve devices and front-end mapping addresses for future configuration and masking operations. This feature reserves the devices/addresses you plan on using, verifies that no one else has reserved the resources, and releases 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 specifying 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 not released explicitly by the user.

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 CLI Command Reference.

Reserve devices

Syntax

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
```

Note

When reserving devices, note the returned reserve ID for subsequent processing.

Options

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, and the file lists configuration change commands.
- To reserve devices for non-specific operations, use the following file syntax:

  ```
  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.

Commit changes on reserved devices

Description

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.

Syntax

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]]]
```

Examples

To commit the changes in the command file `delete.cmd`, enter:

```
symconfigure -sid 3241 -file delete.cmd commit -reserve_id 5
```

View reserved devices on an array

Syntax

To view the devices reserved on an array, use the following command:

```
symconfigure -sid SymmID -reserved list
```

View details on reservation ID

Syntax

To view details on a specific reservation ID, use the following command:

```
symconfigure -reserve_id ResvID show
```

Options

ResvID

The device reservation ID.

Release reserved devices

Description

When releasing device reservations, supply the appropriate reserve ID. Performing a configuration change on reserved devices will not release them. Releasing a reservation is an independent step.

Syntax

To release reserved devices, use the following form:

```
symconfigure -sid SymmID [-noprompt]
 -reserve_id ResvID[,ResvID[,ResvID]] release
```

Options

ResvID

Specifies the device reservation ID.

Examples

To release the set of devices with the ID **5** and **7**, enter:

```
symconfigure -sid 3241 release -reserve_id 5,7
```

Device pool management (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 Virtual Provisioning overview (HYPERMAX OS 5977) on page 332.

Device pool management (Enginuity 5773 - 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 User Guide.
- *SRDF/A DSE pools*, used for SRDF operations, extend 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 User 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 331.

Device pool attributes and rules (Enginuity 5773 - 5876)

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` setting in the options file 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 pools:

- A special Snap pool DEFAULT_POOL exists by default; it 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/A DSE pools

The following attributes and rules apply to SRDF/A DSE pools:

- SRDF/A DSE pools can be associated (shared) with multiple SRDF/A groups.
- A single SRDF/A group can have, at most, four SRDF/A 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.

Create a device pool (Enginuity 5773 - 5876)

Syntax

To create a snap or SRDF/A DSE device pool, use the following form:

```
symconfigure create pool PoolName, type = snap | rdfa_dse
```

Note

The maximum number of device pools per array is 512.

Example

To create a new snap pool named **Finance**, enter:

```
symconfigure create pool Finance, type=snap;
```

Delete a device pool (Enginuity 5773 - 5876)

Description

You can only delete a device pool if it does not have any members.

Syntax

To delete a device pool, use the following form:

```
symconfigure delete pool PoolName, type = snap | rdfa_dse | thin;
```

Example

To delete a snap pool named **Finance**, enter:

```
symconfigure delete pool Finance, type=snap;
```

Add devices to a pool (Enginuity 5773 - 5876)

Description

Only disabled and inactive devices can be added to a pool. While adding the devices, you can specify whether they should be enabled or disabled in the pool.

Syntax

To add devices to a pool, use the following form:

```
symconfigure add dev SymDevName[:SymDevName] to pool PoolName
 type = snap | rdfa_dse | thin [, member_state = ENABLE | DISABLE];
```

Options

member_state

Indicates whether the device(s) being added should be enabled or disabled in the pool.

Example

To add and enable devices from **0001B** to **0001D** to a thin provisioning data pool named **ACT20**, enter:

```
symconfigure add dev 0001B:0001D to pool ACT20, type=thin,
member_state=ENABLE;
```

Remove devices from a pool (Enginuity 5773 - 5876)

Description

Only disabled and inactive devices can be removed from a pool. Once the device is removed from a pool, the device is available to be added to any other pool.

Note

The last DATA device of a pool in a thin tier cannot be removed, even if that pool has no bound devices.

Syntax

To remove devices from a pool, use the following form:

```
symconfigure remove dev SymDevName[:SymDevName] from pool PoolName,
type = snap | rdfa_dse | thin;
```

Example

To remove device **0001E** from a pool named **HR30**, enter:

```
symconfigure remove dev 0001E from pool HR30, type=rdfa_dse;
```

Enable or disable devices in a pool (Enginuity 5773 - 5876)

Description

Devices can be enabled or disabled 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`.

Syntax

To enable or disable a device in a pool, use the following form:

```
symconfigure enable | disable dev SymDevName[:SymDevName]
 in pool PoolName, type = snap | rdfa_dse | thin;
```

Options

PoolName
: Specifies the name of the device pool. It can be from 1 to 12 alphanumeric characters long and include hyphen (-) and underscore (_) characters.

Example

To enable the use of three new DATA devices in a pool named **Finance**, enter:

```
symconfigure enable dev 0001B:0001D in pool Finance, type=thin;
```

Rename a device pool (Enginuity 5773 - 5876)

Syntax

To rename a device pool, use the following form:

```
symconfigure 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:
  - 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.
- A pool cannot be created and renamed in the same session.

Monitor a device pool (Enginuity 5773 - 5876)

Syntax

To monitor usage of a device pool, 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
```

Options

-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 SRDF/A DSE pool.

-thin
Specifies that the pool type is a thin pool.

View a device pool (Enginuity 5773 - 5876)

Example

To display all pools, use the `symcfg list` command with the array ID:

```
symcfg list -sid 397 -pools
```

Sample output

```
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
```

Note the following:

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.

Map devices

Description

The Device Reallocation feature allows for mapping devices to front-end director ports, or mapping a range of devices to consecutive addresses starting from a specified address.

Syntax

To map a device to a director port, use the following command:

```
symconfigure map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNum[, emulation = EmulationType]
[starting][target = ScsiTarget,] lun = ScsiLun[, vbus = FibreVbus
```

Note

Parameters after the `map dev` command can be in any order.

Options

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.

PortNum
Indicates the port number.

Note

For arrays running HYPERMAX OS 5977, the valid range for *PortNum* is extended to 0 to 31.

Examples

To map device **00030** to director **16A**, port **0**, and SCSI target/LUN **0**, **7**, enter in the command file:

```
symconfigure map dev 00030 to dir 16A:0 target=0, lun=7;
```

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:

```
symconfigure 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, 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 the host's device information has been updated, can cause host errors.
- To update the hosts, run the utilities designed for the specific platform as described in Introduce devices to a host on page 287. 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.
- For arrays running Enginuity 5875 or lower, mapping a RecoverPoint-tagged device or an AS/400 device to an FCoE director is only supported.

Obtain list of addresses

Syntax

To obtain a list of used addresses, including the next available address, use the following command:

```
symcfg list -SA all -address -available
```

Map CKD devices (Enginuity 5773 - 5876)

Solutions Enabler supports the mapping or un-mapping of a range of CKD device from an EA/EF port. In addition, it also supports assigning PAV alias addresses for devices already mapped.

Note

The mapping of CKD devices is not supported for arrays running HYPERMAX OS 5977.

Map a range of devices (Enginuity 5773 - 5876)

Syntax

To map a range of device to an EA/EF port, use the following form:

```
symconfigure map dev SymDevName[:SymDevName] to
dirDirectorNum:PortNum, starting
base_address=cu_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:

```
symconfigure map dev SymDevName[:SymDevName] to
dirDirectorNum:PortNum,
copying dirDirectorNum:PortNum
```

Options

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.

Example
To map devices 0001D through 00023 to CU image A, where the next free base address is 20 and the SSID is 140, enter:

```
symconfigure map dev 0001D:00023 to dir 01C:1 starting
base_address=A20 mvs_ssid=140;
```

Copy the mapping of a range of devices (Enginuity 5773 - 5876)

Description
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.

Syntax
To copy a CU image mapping from one port to another, use the following form:

```
symconfigure map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNumcopying dir DirectorNum:PortNum
```

Example
To copy addresses of devices `00010` through `00050` from port `03C:0` to port `03D:0`, enter:

```
symconfigure map dev 00010:00050 to dir 03D:0 copying dir 03C:0;
```

Copy CU image mapping (Enginuity 5773 - 5876)

Syntax
To copy a CU image mapping from one port to another, use the following form:

```
symconfigure map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNumcopying dir DirectorNum:PortNum
```

Copy mapping of device range (Enginuity 5773 - 5876)

Syntax
To copy addresses of devices `00010` through `00050` from port `03C:0` to port `03D:0`, enter:

```
symconfigure map dev 00010:00050 to dir 03D:0 copying dir 03C:0;
```

Assign PAV alias addresses to mapped devices (Enginuity 5773 - 5876)

Description
When assigning PAV alias addresses to mapped devices, the aliases are propagated to all director ports to which the devices are mapped. Devices within the range that are not mapped are 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.

Syntax
To assign a PAV alias address range to a CU image, use the following form:

```
symconfigure add pav alias_range
nnnnn
:
nnnnn

 to mvs_ssid =
 nnn
;
```

To assign PAV alias addresses to devices mapped to EA or EF ports, use the following syntax:

```
symconfigure add pav alias to dev SymDevName[:SymDevName], alias
count=
nnn
```

Note

For these assignments to take effect, run the `add pav alias_range` command first, then run the `add pav alias to dev` command.

Examples
To add a PAV alias range to a CU image using SSID `140`, enter:

```
symconfigure add pav alias_range addr 00080:0009f to mvs_ssid=140;
```

To add the alias `A60` to device `01D`, which is already mapped, enter:

```
symconfigure add pav alias to dev 01D starting alias=A60
```

To add total PAV aliases to device `37a`, enter:

```
symconfigure add pav alias to dev 37a , alias count=32
```

Remove PAV alias addresses from mapped devices (Enginuity 5773 - 5876)

Syntax
To remove PAV alias addresses from devices mapped to EA or EF ports, use the following syntaxy:

To add total PAV aliases to device `37a`, enter:

```
symconfigure remove pav alias from dev SymDevName[:SymDevName], alias
count=
nnn
```

Remove PAV alias addresses from CU image (Enginuity 5773 - 5876)

Syntax
To remove a PAV alias address range from a CU image, use the following syntax:

```
symconfigure remove pav alias_range from mvs_ssid =
nnn
```

Unmap devices

Description

The Device Reallocation feature allows for unmapping devices from front-end director ports.

Unmapping can be from one or all ports. Since all devices with the same SSID must either be mapped or unmapped, specify an SSID when unmapping only some devices in a CU image.

Syntax

To unmap devices from a director port, use the following form:

```
symconfigure unmap dev SymDevName[:SymDevName] from dir
<ALL:ALL | ALL:PortNum | DirectorNum:ALL | DirectorNum:PortNum>
[, emulation = EmulationType]
[, devmask_access = remove | retain];
```

Options

PortNum
Indicates the port number.

> **Note**
>
> For arrays running HYPERMAX OS 5977, the valid range for *PortNum* is extended to 0 to 31.

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.

Restrictions

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.

Unmap devices from EA or EF ports

Syntax

To unmap a range of devices from EA or EF ports, use the following form:

```
 symconfigureunmap dev SymDevName:[SymDevName] from dir
< ALL:ALL | ALL:PortNum | DirectorNum:ALL | DirectorNum:PortNum
[new_ssid = nnn];
```

Options

PortNum
Indicates the port number.

Note

For arrays running HYPERMAX OS 5977, the valid range for *PortNum* is extended to 0 to 31.

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:

```
symconfigure unmap dev 0013B:0013F from dir all:all, new_ssid=0160;
```

I/O activity and unmapping devices

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
  ```

- When unmapping only one path to a multi-pathed device, you may prefer to write disable that path only:

  ```
  symdg -g DgName -SA 16A -p 0 write_disable
  ```

  To 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 the host's device information has been updated, can cause host errors.
  To update the hosts, run the utilities designed for the specific platform as described in Introduce devices to a host on page 287. After the host environment is updated, I/O activity can resume with the array device.

Introduce devices to a host

After reconfiguring an array by moving, deleting, adding, or modifying one or more devices, 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.

Introduce devices to Sun Solaris systems

Syntax

To add or modify devices while online in the Solaris environment, use the following form:

```
drvconfig | symcfg scan
disks
devlinks
```

Note

Follow the instructions in the Solaris documentation to introduce new devices to the host environment.

Introduce devices to HP-UX systems

Syntax

To view mapping change results for HP-UX hosts, use the `ioscan` command in a statement similar to the following:

```
ioscan -fnC disk
```

To define newly connected physical volumes to the HP-UX host system without rebooting it, use the following form:

```
insf -e
```

Note

For more information, refer to the HP 9000 documentation.

Introduce devices to IBM AIX systems

To introduce new devices for AIX hosts, perform the following actions:

Procedure

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.

Introduce devices to HP Tru64 UNIX systems

To introduce new devices for Tru64 UNIX hosts, perform the following actions:

Procedure

1. At the prompt, type:

   ```
   scsimgr -scan_bus bus=BUSNUM
   ```

2. Repeat for each LUN:

3. Write a label to the device you are defining:

```
disklabel -rw rz<lun_letter><unitID> <label>
```

4. Change the ownership on the device to a particular application:

```
chown <owner>:<group> *rz<lun letter><unitID>*
```

5. Follow the host documentation to introduce new devices to the host environment.

Introduce devices to Windows systems

To introduce new or changed devices for Windows hosts, while the system remains online:

Procedure

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.

Configure dynamic SRDF-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 SRDF-capable devices. Dynamic SRDF functionality allows for creating, deleting, and swapping 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, the SRDF device designations of a specified device group can be dynamically swapped 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.

Before dynamic SRDF operations can be performed, the following is required:

- The array must have the dynamic SRDF feature enabled.
- 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.

Enable dynamic SRDF on the device (Enginuity 5773)

Description

Before enabling Non-SRDF devices as dynamic SRDF devices, identify the those devices as follows:

```
symdev list -dynamic
```

Note

For information on using the devices with SRDF commands, refer to the EMC Solutions Enabler SRDF Family CLI User Guide

Syntax

In a command file, use the following syntax:

```
set dev SymDevName[:SymDevName] attribute = dyn_rdf | dyn_rdf1_only |
dyn_rdf2_only
```

Options

dyn_rdf
Using the `dyn_rdf` option provides the most flexibility.

dyn_rdf1_only
This option limits a device to an R1. Using this option revents the ability to perform SRDF swaps.

dyn_rdf2_only
This option limits a device to an R2. Using this option revents the ability to perform SRDF swaps.

Example

Sample command file `set_device_dyn_rdf.cmd`:

```
set dev 00020:0002A attribute=dyn_rdf;
set dev 00040:00045 attribute=dyn_rdf;
set dev 00056:0005A attribute=dyn_rdf;
```

To commit changes:

```
symconfigure -sid 5605 -file set_device_dyn_rdf.cmd -noprompt -v
commit
```

View devices available for use in dynamic SRDF

Description

Before creating a command file to enable dynamic SRDF on devices, identify the non-SRDF devices that you want available for use as dynamic SRDF devices.

Syntax

To view the set of devices currently available for use in dynamic SRDF operations:

```
symdev list -dynamic
```

Commit enabling dynamic SRDF on the device

Examples

To commit, use the following command:

```
symconfigure -sid 5605 -file set_device_dyn_rdf.cmd -noprompt -v
commit
```

Set SRDF group attributes

Description

SRDF 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. Use the `symrdf` command to set SRDF attributes for arrays running Solutions Enabler. 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. Use the `symrdf` command to set SRDF attributes for arrays running Solutions Enabler.

Swap RA groups

Description

Use the `symrdf` command to set swap devices in an RA group from target to source. 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.

Convert directors (Enginuity 5876 only)

Description

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, change of the director emulation type is supported. This support is for FA and RA director types in a switched SRDF environment. Other director types are not supported.

Note

The system checks for Access Control permission CFGSYM and for an SRDF license.

Syntax

Use the `convert dir` option with the `symconfigure` command to convert an FA director to an RF director, and vice versa, as follows:

```
symconfigure convert dir DirNum to type = <RF | FA>;
```

Example

To convert director `7H` to type `RF`, enter:

```
symconfigure convert dir 7H to type = RF;
```

Note

After the `convert director` request executes successfully, issue a `symcfg list` to verify the new director types.

Restrictions when converting directors (Enginuity 5876 only)

FA to RF restrictions

- 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, conversion from FA to RF is possible without SRDF set up on the array.
- The FA director cannot have mapped devices.
- The FA/RF conversion must be a standalone request. Other commands (such as unmapping devices) cannot be executed together with 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 more connectivity is

needed, 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.

The following emulation types support associating (assigning) unused ports to front-end emulations and disassociating (freeing) them:

- FA
- RF

Arrays running HYPERMAX OS 5977 do not support association of ports to EF (FICON) emulations.

Associate ports to director emulations (HYPERMAX OS 5977)

Description

Use the `symconfigure associate port` command to associate one or more physical ports to a director emulation type in the specified director board.

Syntax

To associate a port to a director, use the following syntax:

```
symconfigure associate port <port_num>[,<port_num>. . .] to dir
<dir_num>;
```

Options

dir_num
The director number of the emulation.

Example

To associate ports `2` and `4` to emulation type FA on director `7E`,enter:

```
symconfigure associate port 2,4 to dir 7E ;
```

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. To display free ports, use the CLI command `symcfg list -sid xxx -port -free`.
- Associating ports to RE and EF emulations is not supported.
- Virtual ports cannot 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.

Verify port status after association

Description

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.

Examples

```
symcfg list -dir ALL -sid 064
```

Sample output

```
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
```

Disassociate ports from director emulations (HYPERMAX OS 5977)

Description

Use the `symconfigure disassociate port` command to disassociate one or more physical ports from an emulation type in the specified director board.

Syntax

To disassociate a port from a director type, use the following syntax:

```
symconfigure disassociate port <port_num>[,<port_num>. . .] from dir
<dir_num>;
```

Options

dir_num
The director number of the emulation.

Example

To disassociate ports `1` and `4` from emulation type FA on director `7E`, enter:

```
symconfigure disassociate port 1,4 from dir 7E ;
```

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 cannot be associated or disassociated from an emulation.
- Passthru ports cannot 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.

Verify port status after disassociation

Description

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.

Examples

```
symcfg list -port -free -sid 064
```

Sample output

```
Symmetrix ID: 000197100064
              Flags     Speed
  Slot  Port  FCISDRE   Gb/sec  Status
  ----  ----  -------   ------  -------

     1     4  ...Y...        1  Powered
     1    15  ...Y..Y        1  Powered
     2    24  .Y.....        1  Powered
     2    25  .Y.....        1  Powered
     2    30  Y....Y.        1  Powered
     2    31  Y....Y.        1  Powered

Legend:
  Flags:
    (F)A       : Y = Yes,  . = No
```

```
    F(C)OE      : Y = Yes,  . = No
    F(I)CON     : Y = Yes,  . = No
    (S)E        : Y = Yes,  . = No
    (D)X        : Y = Yes,  . = No
    (R)F        : Y = Yes,  . = No
    R(E)        : Y = Yes,  . = No
```

Set port characteristics

Syntax

To set the port characteristics of a specified director, use the following syntax:

Note

When setting port attributes, it is recommended that you temporarily suspend I/O activity to the effected ports during this operation.

```
symconfigure 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];
```

Restriction

This command cannot be used to set port characteristics for iSCSI physical front end ports and iSCSI target virtual ports. Use the modify iscsi_target on page 311 command.

Options

FlagName

A SCSI or fibre port flag. Possible values for the SCSI protocol flags are in Table 35 on page 297, and the values for the fibre protocol flags are in Table 36 on page 299.

NOTICE

Incorrectly changing the port flags can render the 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.

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). Valid values are 0 through 125. (Hard Addressing must be enabled.) Not applicable for Gig-E ports.

hostname
The 12-character hostname.

Example

To turn on the write protect access logix (**ACLX**) for director 7E, port 0, enter:

```
symconfigure set port 7e:0 ACLX=enable;
```

SCSI protocol flags

Table 35 SCSI protocol port flags

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.)

Table 35 SCSI protocol port flags (continued)

SCSI protocol flags	Description
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 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.)
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.

Table 35 SCSI protocol port flags (continued)

SCSI protocol flags	Description
	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[b]	When enabled, this flag enables SCSI synchronous negotiations. (Default is enabled, and currently, you cannot change this flag.)
Tagged_Commands[b]	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. Not supported in 5874 or higher.

Fibre protocol flags

The following table lists the Fibre protocol flags and their descriptions.

Table 36 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

Table 36 Fibre protocol port flags (continued)

Fibre protocol flags	Description
	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. **Note** This flag is not supported on arrays running HYPERMAX OS 5977. Setting this flag will return a feature not supported error.
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 lower.
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. Not supported for Enginuity 5874 or higher.

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, set the `Show_ACLX_Device` attribute on front end ports to ENABLE or DISABLE, respectively. By default all ACLX enabled ports will have the `Show_ACLX_Device` attribute disabled.

Example
To make the ACLX device visible, enter:

```
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.
```

Report flag details

Description

To report the new SHOW_ACLX_DEVICE port attribute configured on an array port, use either the **list -fa -v** or the list -fa -detail command. The following is an example output reporting the port detail for an ACLX device:

Note

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.

Example

To list port detail for an ACLX device, enter:

```
symcfg -sid 064  list -port -fa 7e -detail
```

The following is an example output reporting the port detail for an ACLX device:

Sample output

```
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
```

CHAPTER 8

Managing multiple iSCSI targets

This chapter describes how to manage multiple iSCSI targets and IP addresses on SE emulation.

- Manage multiple ISCSI targets 304
- Report iSCSI target information 321
- Expanded port group management for iSCSI targets 324
- Report iSCSI target port information 326

Manage multiple ISCSI targets

VMAX 100K, 200K, and 400K arrays running HYPERMAX OS 5977 Q3 2015 SR or higher allow multiple iSCSI targets (IQNs) and IP addresses on SE director emulations. Solutions Enabler, using the `symconfigure` command, provides support for managing (create, modify and delete) IP interfaces and iSCSI targets, and allows attaching and detaching IP interfaces to and from the iSCSI targets. In addition, static IP routes can be added, for routing packets going out of IP interfaces configured on SE director emulations.

The following set of rules and restrictions apply to the management of IP interfaces, iSCSI targets and IP routes:

- Each SE director port supports up to 64 IP interfaces. The maximum number of IP interfaces per SE director emulation is 1000.
- On an SE director emulation, each IP interface is uniquely identified by its IP address/network_id combination. A vlan ID (0 - 4094) must also be configured on this interface, and it must be unique across all IP interfaces defined on the same physical port.
- Each IP interface can be configured with either IPv4 or IPv6 IP address, but not both.
- The maximum number of iSCSI targets per SE director emulation is 256.
- Each iSCSI target must be assigned a network ID, and can be attached to up to 8 IP interfaces on the same SE director emulation. Each of the attached IP interfaces must have the same network ID as the iSCSI target.
- An iSCSI target is uniquely identified by its IQN (iSCSI target name, ASCII string in IQN format). The IQN can be user specified, otherwise HYPERMAX OS generates a globally-unique IQN when the iSCSI target is created. An iSCSI target can also be uniquely identified by a director emulation number and virtual port number, the iSCSI target virtual port. HYPERMAX OS assigns a iSCSI virtual port (0 - 255) when each iSCSI target is created.

  Note

  iSCSI targets are used as endpoints to the iSCSI protocol; an iSCSI virtual port is different from the SE emulation physical ports.

- The maximum number of IP routes (IPv4 + IPv6) per director emulation is 1024.

iSCSI Configuration sessions

Configuration changes in `symconfigure` sessions are done in the order that they are given. If an IP Interface or iSCSI target is created during a configuration session these can be modified or used by other configuration changes that are called for later in the same session. Therefore, if an IP Interface or iSCSI target is deleted during a configuration session, attempts to modify or use these by other configuration changes that are called for later in the same session will fail.

Arrays without embedded management and only iSCSI front-end emulations

Special operating rules exist for VMAX 100K, 200K, and 400K arrays that contain only iSCSI front-end emulations (SEs) and are running without Embedded Management. On these systems, during the installation process, HYPERMAX OS creates a bootstrap iSCSI target on one of the SE emulations, an IP Interface object (with a pre-defined IP address) on one of that SE's physical ports, and maps the ACLX device to that iSCSI Target. This provides control access through that IP address to all hosts and allows the initial provisioning of control hosts and creation of masking views. HYPERMAX OS 5977 or higher, blocks all host I/Os while this bootstrap iSCSI target exists. To enable host I/Os,

delete this bootstrap iSCSI target after provisioning is established. Solutions Enabler provides the following features to support the iSCSI bootstrap target:

- A filter option is added to the `symcfg list` command that displays properties of the iSCSI bootstrap target on the array.
- With the exception of detaching from its IP Interface, all operations for the iSCSI bootstrap target are disabled. It cannot be added to a port group or a masking view, and the configuration cannot be modified (rename, modify or attach to another IP Interface).

Create IP interface on SE director port

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

Use the `create_ip_interface` command to create an IP interface on a SE director port. Only one IP address is allowed and it must be either IPv4 or IPv6. This command requires an IP address, a subnetwork prefix, a network ID, and a vlan ID.

Syntax

To create an IP interface, use the following syntax:

```
create ip_interface dir <director_num> port <port_number>,
       ip_address=<IPaddress>, ip_prefix=<ip_prefix>,
       network_id = <network_id>, vlanid=<vlanid>
       [, mtu = <MTU>];
```

Options

director_num
: SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

port_num
: Port number on a director where the IP interface is created. Valid values are 0-31.

IPaddress
: For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

ip-prefix
: For IPv4 prefix value is 1-30. For IPv6 prefix value is 1-128.

network_id
: Valid values are 1 - 16383.

vlanid
: Valid values are 0 - 4094.

MTU
: Maximum transmission unit. Valid values are 1500 - 9000. The default value is 1500.

Examples

Create command file `create_ip_interface.cmd`:

```
create ip_interface dir 7E port 2, ip_address = 111.111.111.123,
             ip_prefix = 24, network_id = 10, vlanid = 10
```

To commit command file, enter:

```
symconfigure -sid 230 -file create_ip_interface.cmd commit

Execute a symconfigure operation for symmetrix '00019710230' (y/
[n]) ? y
```

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

    Establishing a configuration change
session..............Established.
    Processing symmetrix 000197100230
    {
      create ip_interface dir 7E port 2, ip_address = 111.111.111.123,
        ip_prefix = 24, network_id = 10, vlanid = 10;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director must be a SE director emulation.
- Port must be configured to the director emulation.
- IP address syntax must be valid syntax and the IP prefix must be in the valid range.
- Network ID must be in the valid range.
- Vlan ID must be in the valid range.
- MTU must be in the valid range.
- Maximum IP interfaces per SE director emulation is 1000.
- Maximum IP interface per SE director physical port is 64.
- Vlan ID must be unique to the IP Interface for the specified SE director physical port.
- Subnet mask of the IP Interface must be unique on a network ID within the specified SE director emulation.
- IP address must be unique within the same network ID for the specified SE director emulation.

Modify IP interface on SE director emulation

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax
To modify an IP interface, use the following syntax:

```
modify ip_interface dir <director_num>,
       ip_address = <IPaddress>, network_id = <network_id>
       [, new_network_id = <network_id>]
       [, new_ip_adddres = <IPaddress>]
       [, ip_prefix = <ip_prefix>]
       [, mtu = <mtu>];
```

Options

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

IPaddress

For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

ip-prefix
For IPv4 prefix value is 1-30. For IPv6 prefix value is 1-128.

network_id
Valid values are 1 - 16383.

vlanid
Valid values are 0 - 4094.

MTU
Maximum transmission unit. Valid values are 1500 - 9000. The default value is 1500.

Examples

Create command file `modify_ip_interface.cmd`:

```
modify ip_interface dir 7E, ip_address= 111.111.111.123, network_id =
10, ip_prefix = 24;
```

To commit command file, enter:

```
symconfigure -sid 230 -file modify_ip_interface.cmd commit

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
    {
      modify ip_interface dir 7E , ip_address= 111.111.111.123,
        network_id = 10, ip_prefix = 24;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Director must be a SE director emulation.
- Port must be configured to the director emulation.
- IP address syntax must be valid syntax and the ip prefix must be in the valid range.
- Network ID must be in the valid range.
- MTU must be in the valid range.
- IP address and network ID combination must exist for the director emulation.
- IP address or ip prefix must result in a unique subnet mask across the array.
- IP interface cannot be attached to an iSCSI target.
- IP address or network ID must result in a unique IP address/network ID combination across the array.

- The IP address and network_id together must be globally unique across the entire array.
- Command will fail if all properties are the same as those being changed.

Delete IP interface on SE director emulation

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To delete an IP interface, use the following syntax:

```
delete ip_interface dir <director_num>, ip_address = <IPaddress>,
network_id = <network_id>;
```

Options

IPaddress
For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

network_id
Valid values are 1 - 16383.

Examples

Create command file `delete_ip_interface.cmd`:

```
delete ip_interface dir 7E, ip_address= 111.111.111.123, network_id =
10;
```

To commit command file, enter:

```
symconfigure -sid 230 -file delete_ip_interface.cmd commit

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
    {
      delete ip_interface dir 7E, ip_address= 111.111.111.123,
        network_id = 10;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Director must be a SE director emulation.

- IP address syntax must be valid syntax.
- IP address must exist for the director emulation.
- Network ID must be in the valid range.
- IP interface cannot be attached to an iSCSI target.

Create iSCSI target on SE director emulation

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

Use the `create_iscsi_tgt` command to create an iSCSI target on a SE director emulation. A network ID must be specified, and the IQN (iSCSI qualified name) can either be specified, or if not specified, it is auto-generated by the array. Newly created iSCSI targets are automatically assigned an iSCSI virtual port number (0-255), local to the SE director emulation, and are used to uniquely identifiy the iSCSI target. These virtual ports are used as endpoints to the iSCSI protocol, and are different from physical ports that are mapped to a SE emulation.

Optional settings include IP address, SCSI flags, and the TCP port.

Syntax

To create iSCSI target, use the following syntax:

```
create iscsi_tgt dir <director_num>,
      network_id = <network_id>
      set_default_flags = <ENABLE | DISABLE>]
      [, iqn = <IQN>] [, ip_address = <IPaddress> [,...]]
      [, flag_name = ENABLE [,...]]
      [, tcp_port=<tcp_port>];
```

Options

director_num
: SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

network_id
: Valid values are 1 - 16383.

set_default_flags
: Specifies whether to use iSCSI target default flags (flags listed below).

IQN
: iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

IPaddress
: For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

tcp_port
: Valid values are 1 - 65535. Default value is 3260.

flag_name
: Valid flag values are:

  - SOFT_RESET
  - ENVIRON_SET
  - DISABLE_q_RESET_ON_UA
  - AVOID_RESET_BROADCAST
  - SCSI_3 (DEFAULT is ENABLED)

- SPC2_PROTOCOL_VERSION (DEFAULT is ENABLED)
- ISID_PROTECTED
- SCSI_SUPPORT1 (DEFAULT is ENABLED)
- Volume_Set_Addressing
- OpenVMS

Examples

Create command file `iscsi_tgt_create.cmd`, with IQN specified:

```
create iscsi_tgt dir 7E, iqn = iqn.2013-06.com.emc:sn.11111111,
 network_id = 10, ip_address= 111.111.111.123;
```

To commit command file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_create.cmd commit

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 iscsi_tgt dir 7E, iqn = iqn.2013-06.com.emc:sn.11111111,
        network_id = 10, ip_address= 111.111.111.123;
    }

    Performing Access checks.................................Allowed.
     . . .
      Created IQN : iqn.2013-06.com.emc:sn.11111111

    Committing configuration
changes.......................Committed.
    Terminating the configuration change session.............Done.

The configuration change session has successfully completed.
```

Create command file `iscsi_tgt_create.cmd`, without IQN specified:

```
create iscsi_tgt dir 7E, network_id = 10, ip_address= 111.111.111.123;
```

Note

IQN is auto-generated when the command file is committed.

To commit command file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_create.cmd commit

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 iscsi_tgt dir 7E, network_id = 10,
        ip_address= 111.111.111.123;
    }
```

```
    Performing Access checks.................................Allowed.
     . . .
      Created IQN : iqn.1992-04.com.emc:
600009700bbf824907fd019f00000001

    Committing configuration
changes.........................Committed.
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Director must be a SE director emulation.
- Maximum iSCSI targets per SE director emulation is 1000.
- IP address syntax must be valid syntax.
- IQN must be unique to the array.
- IP address must exist in the director emulation.
- Network ID of the IP interface must match the network ID of the iSCSI target.
- Network ID must be a valid value.
- Maximum IP addresses per iSCSI target is 8.
- ACLX flag cannot be set or modified on an iSCSI target. This flag is always enabled.
- SHOW_ACLX_DEVICE flag cannot be set or modified on an iSCSI target. This flag is always enabled for bootstrap iSCSI targets and always disabled for user-created iSCSI targets.

Modify iSCSI target

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

Use the `modify iscsi_tgt` command to modify scsi port attributes and the TCP port value of the iSCSI target.

Syntax

To modify an iSCSI target, use the following syntax:

```
modify iscsi_tgt
       <[iqn = <IQN>] | [iscsi_dirport =
<director_num>:<port_number>]>
       [, flag_name=ENABLE | DISABLE [,...]][, tcp_port=<tcp_port>]
       [, network_id=<network_id>];
```

Options

director_num
: SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

port_num
: Port number on a director where the IP interface is created. Valid values are 0-31.

IQN

iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

network_id
Valid values are 1 - 16383.

tcp_port
Valid values are 1 - 65535. Default value is 3260.

flag_name
Valid flag values are:

- SOFT_RESET
- ENVIRON_SET
- DISABLE_Q_RESET_ON_UA
- AVOID_RESET_BROADCAST
- SCSI_3 (DEFAULT is ENABLED)
- SPC2_PROTOCOL_VERSION (DEFAULT is ENABLED)
- ISID_PROTECTED
- SCSI_SUPPORT1 (DEFAULT is ENABLED)
- Volume_Set_Addressing
- OpenVMS

Examples

To enable scsi_3 flag, create command file `iscsi_tgt_mod.cmd`:

```
modify iscsi_tgt, iqn = iqn.2013-06.com.emc:sn.11111111, scsi_3=
enable;
```

To commit command file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_mod.cmd commit

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
    {
        modify iscsi_tgt, iqn = iqn.2013-06.com.emc:sn.11111111,
        scsi_3= enable;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director must be a SE director emulation.

- iSCSi target must exist but cannot be the bootstrap target.
- If modifying SCSI flags, iSCSI target must be in an offline state.
- Specified SCSI flag must be a valid flag.
- Network ID must be a valid value.
- Network ID of the IP interface must match the network ID of the iSCSI target.
- If modifying network ID, iSCSI target must be in an offline state.
- iSCSI target cannot be attached to an IP interface.
- ACLX flag cannot be set or modified on an iSCSI target. This flag is always enabled.
- SHOW_ACLX_DEVICE flag cannot be set or modified on an iSCSI target. This flag is always enabled for bootstrap iSCSI targets and always disabled for user-created iSCSI targets.

Delete iSCSI target

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description
To delete an iSCSI target either specify either the IQN (iSCSI qualified name) or the iSCSI virtual port equivalent as the iSCSI target.

Syntax
To delete an iSCSI target, use the following syntax:

```
delete iscsi_tgt <[iqn = <IQN>] | [iscsi_dirport =
<director_num>:<port_number>]>;
```

Options

IQN
iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

port_num
Port number on a director where the IP interface is created. Valid values are 0-31.

Examples
Create command file `iscsi_tgt_delete.cmd`:

```
delete iscsi_tgt iscsi_dirport = 7E:0 ;
```

To commit command file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_delete.cmd commit

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
    {
      delete iscsi_tgt, iqn = iqn.2013-06.com.emc:sn.11111111;
```

```
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director must be a SE director emulation.
- iSCSi target must exist.
- The iSCSI target cannot be the bootstrap target if no masking views exist on the array.
- iSCSI target must be in an Offline state.
- iSCSI target cannot be in a port group.

Rename iSCSI target

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

To rename an iSCSI target specify either the IQN (iSCSI qualified name) or the iSCSI virtual port equivalent as the iSCSI target, and specify the new IQN.

Syntax

To rename an iSCSI target, use the following syntax:

```
rename iscsi_tgt <[iqn = <IQN>] | [iscsi_dirport =
<director_num>:<port_number>]>
        to new_iqn = <IQN>;
```

Options

IQN
: iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

director_num
: SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation

port_num
: Port number on a director where the IP interface is created. Valid values are 0-31.

Examples

Create command file `iscsi_tgt_rename.cmd`:

```
rename iscsi_tgt iqn = iqn.2013-06.com.emc:sn.11111111,
          to new_iqn = iqn.2013-12.com.emc:sn.11111111
```

To commit command file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_rename.cmd commit
```

```
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
    {
      rename iscsi_tgt iqn = iqn.2013-06.com.emc:sn.11111111,
        to new_iqn = iqn.2013-12.com.emc:sn.11111111 ;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director port must be a SE director emulation.
- Existing and new IQN naming rules apply. See Options on page 314.
- iSCSi target must exist.
- The iSCSI target cannot be the bootstrap target.
- New IQN must be unique.

Attach IP interface to iSCSI target

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

To attach an IP interface to an iSCSI target, specify either the IQN (iSCSI qualified name) or the iSCSI virtual port equivalent as the iSCSI target.

Syntax

To attach an IP address, use the following syntax:

```
attach ip_interface ip_address = <IPaddress>, to iscsi_tgt
    <[iqn = <IQN>] | [iscsi_dirport = <director_num>:<port_number>]>;
```

Options

IPaddress
: For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

IQN
: iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

director_num
: SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

port_num
: Port number on a director where the IP interface is created. Valid values are 0-31.

Examples

Create command file `iscsi_tgt_attach.cmd` using the IQN:

```
attach ip_interface ip_address = 10.10.10.1, to iqn = iqn.
2013-06.com.emc:sn.11111111
```

Create command file `iscsi_tgt_attach.cmd` using the iSCSI virtual port:

```
attach ip_interface ip_address = 10.10.10.1, to iscsi_dirport = 7E:0;
```

To commit file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_attach.cmd commit

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
    {
      attach ip_interface ip_address = 10.10.10.1, to
        iqn = iqn.2013-06.com.emc:sn.11111111;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director port must be a SE director emulation.
- iSCSi target must exist.
- IP address must exist on the director emulation of the iSCSI target's with the same network ID.
- IP address must have no assignment to any iSCSI target on the director emulation.

Detach IP interface from iSCSI target

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

To detach an IP interface from an iSCSI target, specify either the IQN (iSCSI qualified name) or the iSCSI virtual port equivalent of the iSCSI target.

Syntax

To detach an IP address, use the following syntax:

```
detach ip_interface ip_address = <IPaddress> from iscsi_tgt
       <[iqn = <IQN>] | [iscsi_dirport =
<director_num>:<port_number>]>;
```

Options

IPaddress
For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

IQN
iSCSI qualified name. Must start with either "iqn." or "eui." strings and must include alphanumeric characters, colons, dashes, and periods. Maximum length is 255 characters.

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

port_num
Port number on a director where the IP interface is created. Valid values are 0-31.

Examples

Create command file `iscsi_tgt_detach.cmd` using the IQN:

```
detach ip_interface ip_address = 10.10.10.1, from iscsi_tgt iqn = iqn.
2013-06.com.emc:sn.11111111 ;
```

Create command file `iscsi_tgt_detach.cmd` using the iSCSI virtual port:

```
detach ip_interface ip_address = 10.10.10.1, from iscsi_tgt
iscsi_dirport = 7E:0;
```

To commit file, enter:

```
symconfigure -sid 230 -file iscsi_tgt_detach.cmd commit

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
    {
      detach ip_interface ip_address = 10.10.10.1, from
        iscsi_tgt iqn = iqn.2013-06.com.emc:sn.11111111;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director port must be a SE director emulation.
- iSCSi target must exist.
- IP address must be attached to the iSCSI target.

Add IP route to SE director emulation

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To add IP route, use the following syntax:

```
add ip_route dir <director_num>,
       ip_address = <IPaddress>, ip_prefix =
<ip_prefix>,
       gateway = <IPaddress>
       [, network_id = <network_id>];
```

Options

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

IPaddress
For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

ip-prefix
For IPv4 prefix length is 1-32 characters. For IPv6 length is 1-128 characters.

network_id
Valid values are 1 - 16383.

Examples

Create command file `add_ip_route.cmd`:

```
add ip_route dir 7E, ip_address = 10.10.10.0, ip_prefix = 24, gateway
= 10.10.9.1, network_id = 10;
```

To commit command file, enter:

```
symconfigure -sid 230 -file add_ip_route.cmd commit

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
    {
      add ip_route dir 7E,ip_address = 10.10.10.0, ip_prefix = 24,
        gateway = 10.10.9.1 , network_id = 10;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin

- For specified director port, director port must be a SE director emulation.
- IPv4 and IPv6 routes can be added separately, but not in the same command.
- A default route through a gateway can be specified as follows:
  - For IPv4 — *ip_address* (destination) 0.0.0.0 with *ip_prefix* =0.
  - For IPv6 — *ip_address* (destination) ::, with *ip_prefix* =0.
- Only one default gateway per director emulation is allowed.
- Maximum IP routes per SE director board is 1024.

Remove IP route from SE director emulation

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To remove IP route, use the following syntax:

```
remove ip_route dir <director_num>,
       ip_address = <Ipaddress>
       [, network_id = <network_id>];
```

Options

director_num
SE director where the IP interface is created. Valid values are 1 - 128 and must represent an SE director emulation.

IPaddress
For IPv4, must be specified with dotted decimal notation format. For IPv6 must be specified in colon-hexadecimal format.

network_id
Valid values are 1 - 16383.

Examples

Create command file `remove_ip_route.cmd`:

```
remove ip_route dir 7E, ip_address = 1:1:2:: , network_id = 10;
```

To commit command file, enter:

```
symconfigure -sid 230 -file remove_ip_route.cmd commit

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
    {
      remove ip_route dir 7E, ip_address = 1:1:2:: , network_id =
10;
    }

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

The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- For specified director port, director port must be a SE director emulation.
- The IP route must exist.

Example iSCSI configuration

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Configuring a Single IP interface and iSCSI target
The following example configures an iSCSI target on an SE director that can be used for provisioning LUNs to an IPv4 accessible host:

- Create an iSCSI target with a given IQN and network_id of 10 on an SE director emulation.
- Create an IP interface on a physical port associated to the same SE director emulation by giving it an IPv4 address and network prefix of 13.17.253.21/24, a vlan ID of 0 (must be unique to the port), and the network_id of 10.
- Attach to the iSCSI target the configured IP interface using its IPv4 address.
- Add a default IPv4 route (0.0.0.0) on the same SE director using the configured IPv4 interface for the outgoing IPv4 packets.

Configuring a Second IP interface to the same iSCSI Target
The following example configures a second IP interface to the existing iSCSI target (configured above) on the same SE director:

- Create a second IP interface on a physical port associated to the same SE director emulation by giving it a different IPv4 address and network prefix of 13.17.255.23/24, a vlan ID of 1 (must be unique to the port), and the same network_id of 10.
- Attach to the iSCSI target the configured IP interface using its IPv4 address.
- Add a default IPv4 route (0.0.0.0) on the same SE director using the configured IPv4 interface for the outgoing IPv4 packets.

The iSCSI target can now be used to make devices visible to hosts by adding it to a port group.

Symconfigure command restrictions for iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

The following is a list of `symconfigure` command behavior for iSCSI targets:

- `symconfigure set port` — Does not support setting port characteristics for iSCSI physical front end ports and iSCSI target virtual ports. Use `modify iscsi_tgt` command to set flags on iSCSI target virtual ports.
- `symconfigure set port copying port` — Doesnot support copying port characteristics for iSCSI physical front end ports and iSCSI target virtual ports.
- `symconfigure associate port` — Supports association of one or more free iSCSI physical ports to SE director emulation. The following restrictions for this command are:

- If any of the specified ports is already associated with an emulation, the operation fails and displays error.
- If any of the specified ports are non iSCSI physical ports and the director emulation specified is SE, the operation fails and displays error.
- If the specified ports are iSCSI physical ports and the director emulation specified is anything other than SE, the operation fails and displays error.
- Association of ports to director emulation configures the ports in an *Offline* state. Ports states must be explicitly changed to *Online* to make them usable.

- `symconfigure disassociate port` — Supports disassociation of one or more iSCSI physical ports from SE director emulation. The following restrictions for this command are:
  - If any of the specified ports is not associated with the specified director, the operation fails and displays error.
  - Ports must be in *Offline* state.
  - Ports cannot have any devices mapped to it.
  - If the specified director is running an SE emulation and one of the specified iSCSI physical port is in a port group, the operation fails and displays error.
- `symconfigure convert dir` — Does not support conversion of SE director to a different director emulation type and vice versa.
- `symconfigure preview` — Can only provide the best effort checks described in the symconfigure IP and iSCSI target management operations.

Set online or offline state for iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description
Some iSCSI target operations require the associated port to be an *Offline* state. Once operations are complete the port must be returned to the *Online* state to connect with a host.

Syntax
To set iSCSI target online/offline states, use the following syntax:

```
symcfg -sid <SymmID> [-noprompt] [-v]
    -SE <#> <-p <#> | -iscsi_port <#>>
        online
        offline
```

Report iSCSI target information

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

ISCSI target information is reported using the `symcfg list -ip, list _iscsi_tgt`, and `list -route` commands.

List array IP addresses

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To list the IP addresses configured on an array, use the following syntax:

```
symcfg [-sid <SymmID>] [-offline]
   list -ip [-SE <#|ALL>] [-p <#>] [-by_ip]
   list -ip [-RE <#|ALL>] [-p <#>] [-by_ip]
```

Examples

To list IP addresses configured on SE port `10H` of array `230`, enter:

```
symcfg -sid 230 list –ip -se 10H
```

Sample output

```
Symmetrix ID: 000197100230 (Local)
                                                                    iSCSI
Dir:P  NetId  Vlan   IP Address                                  Mtu  Port
------ -----  ---- ------------------------------------------- ---- -----
10H:00     1     1 10.10.10.1/24                               1500     0
           1     2 10.10.11.1/24                               1500    -
           2     3 3ffe:80c0:22c0:1049:1260:1600:3ce0:1234/64  1500     1
10H:02     1     1 11.11.11.1/24                               1500     0
           4     2 11.11.12.1/24                               1500   128
```

List array iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To list iSCSI target configured on an array, use the following syntax:

```
symcfg [-sid <SymmID>] [-offline]
   list <-SE <#|ALL>>
        <-iscsi_tgt [-iqn <TargetIQN> | -iscsi_port <#> | -
bootstrap]
               [-by_iqn] [-detail]>
```

Options

-iqn *TargetIQN*
: Lists only the specified IQN.

-iscsi port *#*
: Lists only the targets configured on the iscsi virtual port.

-bootstrap
: Lists only the bootstrap iSCSI target.

-by_iqn
: Display sort order is by IQN.

-detail
: Lists details of IP interfaces and scsi port flags settings for each iSCSI target.

Examples

To list the iSCSI targets on SE director `10H` for array `230`, enter:

```
symcfg –sid 230 list –iscsi_tgt -SE 10H
```

Sample output

Note

Use the SYMCLI_FULL_NAME option to display more than 57 characters for the IQN.

```
Symmetrix ID: 000197100230 (Local)


Dir:P   NetId Status    IQN
------- ----- ------- ------------------------------------------------------------
10H:000     1 Online  iqn.1992-04.com.emc:5000097300092190
  ...
10H:255     4 Offline iqn.1992-04.com.emc:5000097300092194
```

List array IP routes

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Syntax

To list IP routes configured on an array, use the following syntax:

```
symcfg [-sid <SymmID>] [-offline]
   list -route [-SE <#>] [ -ipv4 | -ipv6 ] [-v]
```

Options

-ipv4 | -ipv6
Lists only ipv4 or ipv6 routes

Examples

To list –ipv4 route for SE director 10H on array 230, enter:

```
symcfg -sid 230 list -route -SE 10H -ipv4
```

Sample output

```
Symmetrix ID: 000197100230 (Local)
Director Identification: SE-10H

Network Id : 5

Destination                          Gateway
------------------------------------ ------------------------------------
87.215.195.176/29                    0.0.0.0
62.58.50.0/25                        62.58.236.234

Network Id : 6

Destination                          Gateway
------------------------------------ ------------------------------------
192.168.1.0/24                       0.0.0.0
10.0.0.0/8                           0.0.0.0
```

Note

In above table 87.215.195.176/29 means destination ipv4 address is 87.215.195.176 and prefix length = 29 (equivalent to netmask = 255.255.255.248).

Output for –ipv6 route:

```
Symmetrix ID: 000197100230 (Local)

Director Identification: SE-10H

Network Id  : 7

Destination                           Gateway
------------------------------------- ---------------------------------------
::1/128                               ::1
8000:1:1::/64                         4000:1:1::1
```

Expanded symcfg and sympd reporting for iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description
The `symcfg list –address` and `sympd list` commands include the `–iscsi_port #` option to support iSCSI targets.

Syntax
For the `symcfg list –address` command use the following syntax with the `–iscsi_port #` option:

```
symcfg [-sid <SymmID>] [-offline] list [-SE  <# | ALL>] [<-v |
      -port [-detail] [-p <#>]>] [-address [-available]] [-iscsi_port
<#>]
```

For the `sympd list` command use the following syntax with the `–iscsi_port #` option:

```
sympd [-offline] [-sid <SymmID>] [-v]
       list [-SA <#|ALL>] [-p <#>] [-scsi] [-fibre]
            [-escon] [-ficon] [-gige | [-iscsi_port <#>]]
            [-powerpath] [-vcm | -aclx] [-pdevfile] [-cyl]
```

Note

The `–iscsi_port #` option is only supported for arrays running HYPERMAX OS 5977 or higher.

Expanded port group management for iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description
For port group management, the symaccess command includes two options to support iSCSI targets, `–iscsi_dirport` and `–iqn` options. These options are used with the following `symaccess` port group operations:

- create, delete, rename add, remove
- list and show
- set, enable, disable, delete, and list CHAP

Additional symaccess commands that support iSCSI targets but do require any CLI changes are:

- The `symaccess copy -type port` and `copy view` commands that copy a port group with iSCSI ports to a target array only if the target array is running HYPERMAX OS 5977 or higher.
- The `symaccess backup` and `symaccess restore` commands backup and restore port groups with iSCSI targets. These commands also backup and restore a provision view masking record with iSCSI virtual ports.

Syntax

For port group management (create, add, remove, list, set CHAP, enable CHAP and display for iSCSI targets, use the following syntax with the `-iscsi_dirport` and `-iqn`options.

```
symaccess -sid <SymmID> -name <GroupName> -type port
     create
     create -dirport <Dir>:<Port>[,<Dir>:<Port>...]
     create -iscsi_dirport <Dir>:<Port>[,<Dir>:<Port>...]
     create -iqn <TargetIQN>[,<TargetIQN>...]
     delete [-force][-noprompt]
     rename -new_name <NewGroupName>
```

```
symaccess -sid <SymmID> -name <GroupName> -type port
        [-celerra][-rp][-ckd]
     add -dirport <Dir>:<Port>[,<Dir>:<Port>...]
     add -iscsi_dirport <Dir>:<Port>[,<Dir>:<Port>...]
     add -iqn <TargetIQN>[,<TargetIQN>...]
```

```
symaccess -sid <SymmID> -name <GroupName> -type port
        [-ckd][-force][-unmap [-celerra][-rp]]
     remove -dirport <Dir>:<Port>[,<Dir>:<Port>...]
     remove -iscsi_dirport <Dir>:<Port>[,<Dir>:<Port>...]
     remove -iqn <TargetIQN>[,<TargetIQN>...]
```

```
symaccess -sid <SymmID> | -file <backup_filename>
     list -type port [-name <GroupName>] [-detail | -v]
           [-dirport <Dir>:<Port> |
            -iscsi_dirport <Dir>:<Port> |
            -iqn <TargetIQN>]
     show <GroupName> -type port
```

```
symaccess -sid <SymmID>
        <-dirport <Dir>:<Port> |
         -iscsi_dirport <Dir>:<Port> |
          -iqn <TargetIQN>]
     set chap -cred <Credential> -secret <Secret>
```

```
symaccess -sid <SymmID>
        [-dirport <Dir>:<Port> |
         -iscsi_dirport <Dir>:<Port> |
          -iqn <TargetIQN>]
     enable chap
     disable chap
     delete chap
```

```
symaccess -sid <SymmID> | -file <backup_filename>
     list chap [-dirport <Dir>:<Port> |
             -iscsi_dirport <Dir>:<Port> |
            -iqn <TargetIQN>] [-v]
```

Restrictions

- Either the -iscsi_dirport or -iqn is specified in a command, not both.
- For FA director type only the -dirport option is used in a command.
- For SE director type on the -iscsi_dirport is used in a command.

- Fibre channel ports and iSCSI targets virtual ports are not allowed in the same port group.

Report iSCSI target port information

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Port information for iSCSI targets is reported, with expanded displays, using the following symaccess commands:

- `show -type port`
- `show view`
- `list hba`
- `list devinfo`
- `list assignment`
- `list chap`
- `list logins`

Show port group with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

The `symaccess show -type port` command is expanded to show the iSCSI target name of the iSCSI virtual target port.

Examples

```
symaccess show host1083_ports -sid 001 -type port
```

Sample output

Displays the the iSCSI target name of the iSCSI virtual target port under `Director Idenfication`.

```
Symmetrix ID             : 000197100001

Port Group Name          : host1083_ports
Last update time         : 12:57:49 AM on Mon Apr 15,2014

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ---- ------------------------------------------------------
      SE-15G  000 iqn.1992-04.com.emc:sn.11111111
      SE-15G  255 iqn.1992-04.com.emc:sn.11111112
```

Show masking view with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Description

The `symaccess show view` command is expanded to show the iSCSI target name of the iSCSI virtual target port.

Examples

```
symaccess show view host1083_view -sid 001
```

Sample output

Displays the the iSCSI target name of the iSCSI virtual target port under `Director Idenfication`.

```
Symmetrix ID                : 000197100001
Masking View Name           : host1083_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
   Host Initiators
     {
       ISCSI  : iqn.2002-06.com.host1083 [alias: api1083/api1083]
     }
Port Group Name             : host1083_ports
   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ---- --------------------------------------------------
      SE-15G  000 iqn.1992-04.com.emc:sn.11111111
      SE-15G  255 iqn.1992-04.com.emc:sn.11111112
```

List HBAs with iSCSI targets (HYPERMAX OS 5977 or higher)

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symaccess list hba
```

Sample output

Displays iSCSI target virtual port in the `Dir:Port` column.

```
symaccess list hba

Identifier        Physical Device Path              Symmetrix ID  Dir:Port
----------------  --------------------------------  ------------  --------
iqn.2002-06.com*  c1t50060482D52D5F02d0s23          000197100266  07G:128
```

List device information with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symaccess list devinfo –ig my_ig –sid 001
```

Sample output

Displays iSCSI target virtual port in the `Dir:Port` column.

```
Symmetrix ID          : 000197100001

Initiator Group Name  : my_ig
Last update time      : 01:20:37 PM on Fri Apr 18,2014

Group last update time: 01:20:37 PM on Fri Apr 18,2014

  Host Initiators
    {
      ISCSI  : iqn.2002-06.com.host1082 [alias: api1082/api1082]
    }
```

```
Sym                                     Host      Cap
Dev   Dir:Port Physical Device Name     Lun  Attr (MB)   Masking View Name
----- -------- ----------------------- ---- ---- ------ --------------------
0080  07G:128  Not Visible               1        2063 myview2
0081  07G:128  Not Visible               2        2063 myview2
0082  07G:128  Not Visible               4        2063 myview2
0083  07G:128  Not Visible               3        2063 myview2
                                                ------
Total Capacity                                  8252
```

List device assignments with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symaccess list assignments –dev C3:C6 –sid 001
```

List no assignments:

```
symaccess list no_assignments -dirport 15E:310 –sid 001
```

Sample output

With `assignments`, displays iSCSI target virtual port in the `Dir:Port` column.

```
Symmetrix ID          : 000197100001
Sym
Dev     Identifier        Type   Dir:Port
------  ----------------  -----  ----------------
000C3   iqn.2002-06.com*  iSCSI  SE-10G:128
000C4   iqn.2002-06.com*  iSCSI  SE-10G:128
000C5   10000000C9AE1298  FIBRE  FA-7E:000
000C6   -                 -      -
```

Note

With `no_assignments`, SE directors are not displayed.

List CHAP information with iSCSI targets (HYPERMAX OS 5977 or higher)

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symaccess list chap -sid 001
```

Sample output

Displays iSCSI target virtual port in the `Identifier`column and `Director Port` column, and displays the iSCSI target name of the iSCSI target virtual port.

```
Symmetrix ID            : 000197100001

Director Identification : SE-9G
Director Port           : 128
iSCSI Target Name       : iqn.1992-04.com.emc:sn.11121318

Protocol                : CHAP

Identifier                Type   State     Credential
```

```
------------------------  -----  --------  ------------------------
SE-9G:128                 N/A    ENABLED   credential
```

List login information with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symaccess list logins -sid 001
```

Sample output

Displays iSCSI target virtual port in the `Director Port` field under the SE director.

```
Symmetrix ID             : 000197100001

Director Identification : FA-5E
Director Port           : 0

                            User-generated                      Logged On
Identifier       Type  Node Name         Port Name         FCID   In     Fabric
---------------- ----- ----------------------------------- ------ ------ ------
10000000c9ae1298 Fibre 10000000c9ae1298 10000000c9ae1298 2b1b00 Yes    Yes

Director Identification : SE-9G
Director Port           : 128

                            User-generated                      Logged On
Identifier       Type  Node Name         Port Name         FCID   In     Fabric
---------------- ----- ----------------------------------- ------ ------ ------
iqn.2002-06.com* iSCSI api1082           api1082           2b1b00 Yes    Yes
```

List storage group demand with iSCSI targets

This topic applies only to arrays running HYPERMAX OS 5977 or higher.

Examples

```
symsg -sid 584 list -demand -by_port
```

Sample output

Displays iSCSI target virtual port in the `Dir:Port` column.

```
Symmetrix ID:        000197100584

  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)
-------- ----- -------- ------- -------- -------- --- ------- --------
05E:000   NN          0       0     1000        0   0       0    +1000
05E:001   NN          0       0     1000        0   0       0    +1000
09G:128   NN          0       1        -        0   -       0        -
```

Output with -v (verbose) option, displays the iSCSI target virtual port in the `Dir:Port` column, and the `iSCSI Target Name` of the iSCSI target port (or virtual port) and the `WWN Port Name` of fibre channel ports.

```
Symmetrix ID:        000197100584
```

```
Director Identification             : FA-7E
Director Port                       : 0
WWN Port Name                       : 5000097300092150
Port Total Demand (IO/Sec)          :        0
Number of SGs without Limit (IO/Sec):        1

Port Negotiated Speed (MB/Sec)      : N/A
Port Total Demand (MB/Sec)          :     1000
Percent Port Capability (%)         : N/A
Port Excess (MB/Sec)                : N/A
Number of SGs without Limit (MB/Sec):        0

Storage Groups (1)
    {
    ---------------------------------------------
                              Maximum Demand
                            ---------------------
    Name                     MB/Sec  (%)  IO/Sec
    ----------------------  -------- --- --------
    app_psg1                    1000   0  NoLimit
      app_csg1                  1000   0  NoLimit
      app_csg2                NoLimit   0  NoLimit
    }
 . . .
Director Identification             : SE-9G
Director Port                       : 128
iSCSI Target Name                   : iqn.1992-04.com.emc:sn.11121318
Port Total Demand (IO/Sec)          :        0
Number of SGs without Limit (IO/Sec):        1
....
```

CHAPTER 9

Configuring Virtual Provisioning

This chapter describes Virtual Provisioningâ„¢ concepts and how to perform operations using the SYMCLI configuration change component.

- Virtual Provisioning overview 332
- Create DATA devices (Enginuity 5773 through 5876) 334
- Create thin devices (Enginuity 5773 through 5876) 335
- Create thin pools (Enginuity 5773 through 5876) 337
- Perform actions on thin devices (Enginuity 5773 through 5876) 341
- Allocate, reclaim and free space on thin devices 344
- Background operations on thin devices 347
- Drain and rebalance a thin pool (Enginuity 5874 through 5876) 348
- Virtual Pool (VP) compression (Enginuity 5876 only) 353
- Verify pool and device states 355
- Monitor thin pools 355
- View thin pools 356

Virtual Provisioning overview

Virtual Provisioning (VP) 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.

Virtual Provisioning overview (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. Virtual Provisioning presents a large amount of capacity to a host and then consumes 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 VMAX3 Family with HYPERMAX 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.

Use the `symdev`, `symsg` (storage groups) , `symcg` (composite groups), and `symdg` (device groups) commands to perform the following Virtual Provisioning operations for thin devices:

Table 37 VP operations for thin devices

Operation	Reference
Allocate space on thin devices	Allocate, reclaim and free space on thin devices on page 344
Reclaim space on thin devices	Reclaim allocations on thin devices on page 345
Free device allocations	Free all allocations or written tracks (HYPERMAX OS 5977 only) on page 347
Rename a thin pool	Rename thin pools on page 339
Monitor a thin pool	Monitor thin pools on page 355
View a thin pool	View thin pools on page 356

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

Virtual Provisioning overview (Enginuity 5773 through 5876)

For arrays running Enginuity 5773 through 5876, the Virtual Provisioning feature involves managing the following components:

- *Thin devices* (TDEV)
- *DATA devices* (DATADEV)
- *Thin pools*

Thin devices are created with an inflated capacity, and the actual storage space for the data is written to the thin devices is on the DATA devices. When additional storage is needed, more DATA devices are 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.

NOTICE

Not all Virtual Provisioning operations are supported on earlier Enginuity versions (such as 5773). Refer to Supported configuration operations on page 230 for supported operations for Enginuity 5773 through 5876.

Use the `symconfigure` command to create thin devices, DATA devices, and thin pools.

Use the `symdev`, `symsg` (storage groups), `symcg` (composite groups), and `symdg` (device groups) commands to perform and manage binding and unbinding of thin devices:

Note

The `symconfigure` command is no longer used to perform bind, unbind, rebind, allocate, reclaim, and free operations on thin devices and thin pools.

Table 38 VP operations for thin devices

Operation	Reference
Bind, unbind, rebind thin devices to thin pools	Perform actions on thin devices (Enginuity 5773 through 5876) on page 341

Table 38 VP operations for thin devices (continued)

Operation	Reference
Start allocate, stop allocate	Allocate, reclaim and free space on thin devices on page 344
Free, stop free	Free written tracks on thin devices (Enginuity 5874 through 5876) on page 346
Reclaim, stop reclaim	Reclaim allocations on thin devices on page 345

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.

Create DATA devices (Enginuity 5773 through 5876)

Description

DATA devices are similar to SAVE devices, as they are not visible to the host and are 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.

Syntax

To create DATA devices with the `symconfigure` command , use the following syntax:

```
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>]
```

Options

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 (CYL).

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, SRDF, 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.

Create 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, and writes and reads from them devices like regular devices.

The same metadevice commands (`form meta`, `dissolve meta`, `convert meta`) can be used to configure a thin metadevice. The supported thin metadevice features are as follows:

Table 39 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 Supported thin metadevice conversions (Enginuity 5773 - 5876) on page 264.

Create thin devices using a file

Description

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.

Note

For thin metadevices, refer to Configure thin metadevices (Enginuity 5773 - 5876) on page 259

Syntax

To create thin devices, 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]];
```

Options

create dev count
Specifies the number of thin devices you are creating.

size
Specifies the size of each thin device. If MB or GB is not specified, the size defaults to cylinders (CYL).

emulation
Indicates FBA, CKD 3390, or AS/400 D910_099 devices.

Note

Thin CKD devices require Enginuity 5876 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.

Note

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.

Examples
To create thin devices using file `add_new_tdevs.cmd`, enter:

```
symconfigure commit -sid 12345 -file add_new_tdevs.cmd
```

The file `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;
```

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. Valid thin device conversions on page 262 lists the valid conversions.

Create thin pools (Enginuity 5773 through 5876)

Use the `symconfigure` command to create thin pools. Thin pools can be created at the same time as creating the DATA devices. The same naming conventions that apply to SAVE pools and SRDF/A DSE pools apply. However, there is no default thin pool. The maximum number of device pools per array is 512.

Note

When creating thin pools, set up thin pool monitoring with the event daemon (`storevntd`) to allow for monitoring of pool out-of-space conditions that may result in application outages.

Syntax

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>];
```

Options

PoolName
Specifies the name of the device pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-) and underscores (_)

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.

max_subs_percent
The *maximum subscription percent* sets the maximum limit of total capacity that can be bound to the thin pool. This ratio is the total host-perceived capacity of thin devices, divided by physical thin pool capacity.
Valid values are 0 through 65534.

Table 40 Maximum subscription percent capacities

When the value is...	Total capacity of the thin devices bound to a pool cannot exceed...
1	1/100th the actual physical pool capacity
100	100 percent of the actual physical pool capacity
200	Twice the amount of the actual physical pool capacity
65534	65534 percent of the total physical pool capacity

rebalance_variance
Target device utilization variance (in %) 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.
Valid values are 1 through 50. The default value is 1.

max_dev_per_rebalance_scan
Maximum number of devices in a pool on which the rebalancing algorithm will concurrently operate.
Valid values are 2 through 1024. The default value is 256.

pool_resv_cap
Percentage of the capacity of the virtual pool that is 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 are 1 through 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 Set FAST control parameters on page 207.

vp_compression
Enables compression on the thin pool. When a thin pool is enabled for *compression* , the data in the pool can be compressed and uncompressed using a corresponding command at the device or group level. Virtual Pool (VP) compression (Enginuity 5876 only) on page 353 provides more details.

Note

VP compression is only supported on arrays running Enginuity 5876. VP compression of CKD devices is not supported on VMAX 10K arrays.

Thin pool restrictions and rules

The following restrictions and rules 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.

Rename thin pools

Syntax
To rename a thin pool, use the following syntax:

```
rename pool PoolName to NewPoolName
```

Set pool attributes (Enginuity 5773 through 5876)

Description
Pool attributes govern the following:

- Pool subscription and reserved capacity
- Rebalancing
  Pool Rebalancing is an operation in which uneven usage of data devices on a pool is evened out. With Enginuity 5875, the `symconfigure` CLI will allow following rebalancing attributes to be set/reset on any thin pool: *rebalancing variance* and *maximum rebalance scan device range*.

- Compression

Note

Rebalancing and compression can only be performed on thin pools and is not applicable for Snap or SRDF/A DSE pools.

Syntax

To set pool attributes when creating a pool, use the following 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>];
```

To set pool attributes after the thin pool is created, use 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>];
```

Options

max_subs_percent
Specifies the maximum subscription percent for the pool. Create thin pools (Enginuity 5773 through 5876) on page 337 provides more details.

rebalance_variance
Specifies the rebalancing variance for the pool. Create thin pools (Enginuity 5773 through 5876) on page 337 provides more details.

max_dev_per_rebalance_scan
Specifies the maximum rebalance scan device range for the pool. Create thin pools (Enginuity 5773 through 5876) on page 337 provides more details.

pool_resv_cap
Specifies the pool reserved capacity (PRC) for the pool. Create thin pools (Enginuity 5773 through 5876) on page 337 provides more details.

vp_compression
Enables or disables compression for the pool. Create thin pools (Enginuity 5773 through 5876) on page 337 provides more details.

Examples

To enable compression for thin pool `poolTest`, enter:

```
symconfigure -sid 397 -nop -v -cmd "set pool poolTest, type = thin, vp_compression =
ENABLE;" commit
```

Sample output

```
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.
```

Perform actions on thin devices (Enginuity 5773 through 5876)

Description

Refer to esd_r_se_seam_performing_actions_on_thin_devices_(enginuity_5773_through_5876)_1225947 on page 341 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 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.

Syntax

To perform control actions on thin devices, use the following command syntax:

Note

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>`.

For `symdev` command:

```
symdev -sid SymmID [-devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
[,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>]
bind -pool <PoolName>
rebind -pool <PoolName>
unbind
allocate [-persistent | -stop]
free [-stop]
reclaim [-persistent | -stop]
```

For `symdg` command:

```
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
allocate [-persistent | -stop]
free [-stop]
reclaim [-persistent | -stop]
```

For `symsg` command:

```
symsg -sg <SgName> -sid <SymmID> [-i <interval>]
[-c <count>] [-v] [-noprompt] [-rp] [-star]
[-celerra] [-skip -lockid <lockNum>]
bind -pool <PoolName>
rebind -pool <PoolName>
unbind
allocate [-persistent | -stop]
free [-stop]
reclaim [-persistent | -stop]
```

For `symcg` command:

```
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
allocate [-persistent | -stop]
free [-all | -stop]
reclaim [-persistent | -stop]
```

Bind thin devices

Description

A `bind` action associates a thin device to a thin pool.

Examples

To bind thin devices in device group **`MyDg`** to the thin pool **`Mythinpool`**, enter:

```
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`*.

To bind thin devices 1078 - 1081 to the thin pool **`Mythinpool, enter:`**

```
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.

Unbind thin devices

Description

An `unbind` action disassociates a thin device from a thin pool.

Examples

To unbind the thin devices in device group `MyDg`, enter:

```
symdg -sid 234 -nop unbind -dg MyDg
```

The `symsg` and `symcg` commands are the same as `symdg` command using `-sg` *`SgName`* and the `-cg` *`CgName`*.

To unbind thin devices 1078 - 1081, enter:

```
symdev -sid 234 -nop unbind -devs 1078:1081
```

Restrictions

The following are restrictions 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.

Rebind thin devices

Description

A `rebind` action rebinds a thin device to a new pool without moving its data or losing any data. Rebind simply changes the current binding of the thin device to a new pool. Rebind does 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.

Examples

To rebind thin devices in device group **`MyDg`** to the new thin pool **`Mythinpool2`**, enter:

```
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`*.

To bind thin devices 1078 - 1081 to the new thin pool **`Mythinpool2`**.

```
symdev -sid 234 -nop rebind -devs 1078:1081 -pool Mythinpool2
```

Restrictions

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.

Verify device pool binding

Examples

To verify pool binding for device `1078`, enter:

```
symdev show -sid 351 1078
```

Sample Output

```
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.

Allocate, reclaim and free space on thin devices

Description

The `allocate` action allocates storage for specified devices. The `-persistent` option specifies that allocated storage cannot be reclaimed or freed. If any of the tracks specified for persistent allocation are already allocated, the already allocated tracks are 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.

To reclaim allocated storage, use the `reclaim` action with the optional `-persistent` option.

Allocating and reclaiming space on devices is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

The free action is used to free up written tracks on thin devices. The `free` action is supported on Enginuity 5874 through 5876, and the `free -all` action is only supported onHYPERMAX OS 5977.

Actions to allocate, reclaim, and free are performed as background operations.

Allocate space on thin devices

Examples

To start allocation on the thin devices in the device group **myDg**, enter:

```
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*.

To start allocation on thin devices 1078 - 1081, enter:

```
symdev -sid 234 -nop -v allocate -devs 1078:1081 -persistent
```

Restrictions for allocating space on thin devices

The following restrictions apply to the allocate 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.
- 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.

Reclaim allocations on thin devices

Description

Reclaiming persistent allocations frees up tracks that are unwritten or zero-based, even if they are marked as persistent. The `reclaim - persistent` command is the only action that frees up persistent tracks and 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 tracks written with zeros. It will not free up tracks that are marked persistent.

Examples

To start reclaiming space on thin devices in the device group **myDg**, enter:

```
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*.

To start reclaiming space on thin devices 1078 - 1081, enter:

```
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 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 an 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.
```

Free written tracks on thin devices (Enginuity 5874 through 5876)

Description

The `free` action frees up written tracks on thin devices.

Note

For Enginuity 5876 or lower, `free` actions are blocked for any device that is the source or target of a TimeFinder/Clone, Emulation, or Snap session.

Examples

To start freeing up space on thin devices in the device group **myDg**, enter:

```
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.

To start freeing up space on thin devices 1078 - 1081, enter:

```
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.
- Freeing space cannot be performed while a DATA device in the pool is draining.

Free all allocations or written tracks (HYPERMAX OS 5977 only)

Description

On arrays running HYPERMAX OS 5977, use the `free -all` to free up allocations on thin devices. This action frees up allocations even if there is data written to the device. The device must be not mapped or *Not Ready*.

Examples

To start freeing up all allocations on thin devices in device group **myDg**, enter:

```
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*.

To start freeing up space all allocations on thin devices 1078 - 1081, enter:

```
symdev -sid 234 -nop -v free -devs 1078:1081 -all
```

Restrictions for freeing all allocations

The `free -all` command is blocked on any device participating in replication operations.

If the `free -all` operation is running in the background on a device and it is in a *Ready* state, the following operations are not allowed on that device:

- add rdf mirror
- map dev
- convert dev

Background operations on thin devices

Description

Actions to `allocate, free`, or `reclaim` space on thin devices are performed asynchronously in the background and are started with an explicit start action. Solutions

Enabler displays these background tasks as the thin device status, such as *allocating*, *reclaiming*, and so on.

Stop background operations on thin devices

Description
To stop background operations, use the `-stop` option.

Note

Stopping background operations is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

Examples
To stop allocation on thin devices in the device group **myDg**, enter:

```
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*.

To stop allocation on thin device `1078`, enter:

```
symdev -sid 234 -nop -v allocate -devs 1078 -stop
```

Restrictions

- 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 fails.

Drain and rebalance 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:

Table 41 Drain and rebalance command for thin pools

Command	Operation
`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.

Thin pool draining rules and restrictions

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 for thin pool draining
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.

Deactivate devices in a thin pool

Description
The `deactivate` command disables the DATA device without starting a drain. This operation puts the DATA device in either of the following two states:

- *Deactivated* — If the DATA device has any used tracks.
  Deactivated state is determined by a combination of the Inactive state, the Not_Draining state, and the presence of used tracks.
- *Disabled* — If the DATA device has zero used tracks.

On deactivated DATA devices:

- Reads and writes can be performed from previously allocated space.
- No new allocations can be performed.

Note

The `deactivate` command is only supported for Enabled DATA devices. The last DATA device of a pool in a thin tier cannot be deactivated, even if that pool has no bound devices.

Syntax
To deactivate devices in a thin pool, use the following syntax:

```
deactivate dev SymDevName [:SymDevName] in pool PoolName type = thin;
```

Examples
To deactivate device `1078` in pool `MyPool`, enter:

```
symconfigure deactivate dev 1078 in pool MyPool type = thin
```

Start the drain on DATA devices

Description

Syntax

To start the drain on the deactivated DATA devices in a thin pool, use the following syntax:

```
start drain on dev SymDevName [:SymDevName] in pool PoolName type =
thin;
```

Examples

To start the drain on device `1078` in pool **`MyPool`**:

```
symconfigure start drain on dev 1078 in pool MyPool type = thin
```

The drain progress is viewed by listing the DATA devices in the array or using the `symcfg show -pool` command.

Activate devices in a thin pool

Description

The `activate` command is same as the `enable` command, except the `activate` command is not allowed if draining is in progress. After the `activate` command completes, the DATA device goes in the *Enabled* state.

The `activate` command is only supported for deactivated DATA devices with used tracks.

Syntax

To activate devices in a thin pool, use the following syntax:

```
activate dev SymDevName [:SymDevName] in pool PoolName type = thin;
```

Examples

The following command activates device `1078` in pool `MyPool`:

```
symconfigure activate dev 1078 in pool MyPool type = thin
```

Stop the drain on DATA devices

Description

Use 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.

Syntax

To stop the drain on the DATA devices in a thin pool, use the following syntax:

```
stop drain on dev SymDevName [:SymDevName] in pool PoolName type =
thin;
```

Examples

The following command deactivates device `1078` in pool **`MyPool`**:

```
symconfigure stop drain on dev 1078 in pool MyPool type = thin
```

Thin pool rebalancing scenario

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 are drained to the 16 new DATA devices.
   b. Monitor the drain operation using the `symcfg show -pool` command 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.

Start pool rebalancing

Description

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.

The `start` operation does the following:

1. Checks to see if a pool balancing operation is in progress for the specified pool.
2. Initiates pool balancing on the specified pool, if pool balancing is not in progress, While write balancing is in progress, it is possible that data will be moved from deactivated DATA devices but never to deactivated DATA devices. This pool state displays as *Balancing*.

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.

Syntax

To start pool rebalancing on a thin pool, use the following syntax:

```
start balancing on pool PoolName;
```

Examples

To start pool rebalancing on thin pool XUN on array ID **`343`**, enter:

```
symconfigure -sid 343 commit -file balance.cmd
```

The file **balance.cmd** contains:

```
start balancing on pool XUN;
```

Verifying thin pool status

Examples

To check the status of the thin pool XUN on 343, enter:

```
symcfg show -sid 343 -pool XUN -thin
```

Sample output

```
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
```

Stop thin pool rebalancing

Syntax

To stop a rebalancing operation on a thin pool, use the following syntax:

```
stop balancing on pool PoolName;
```

Sample output

When there is no balancing in progress, the output of **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 point in time the used capacity of each DATA device is uniform across the pool. While pool rebalancing is in progress, all pool operations can still occur.

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 is enabled for compression, by setting the compression capability on a VP Pool to enabled. When compression is no longer desired, compression capability 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 compressing 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 has 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 Set pool attributes (Enginuity 5773 through 5876) on page 339.

Enable or disable thin pool VP compression

Description

Use the `symconfigure` command to enable or 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.

Syntax

To set VP compression enable or disable, use the pool attribut the following syntax:

```
set pool <PoolName>,
          type = <thin>,
          <[, vp_compression = <ENABLE | DISABLE>];
```

After a pool is enabled for compression, VP compression or VP uncompression can be started and stopped, as described in Start or stop VP compression on thin devices on page 354.

Start or stop VP compression on thin devices

Syntax

To start or stop VP compression or uncompression for allocations on a thin device, use the following syntax:

Note

Compressing or uncompressing a thin device occurs as a background task.

```
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, using the syntax -file ‹*FileName*›. The file must contain one SymDevName per line.

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 is not 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.

Verify pool and device states

Description

The `symcfg verify` command verifies the states of DATA devices and thin devices, and also determines 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 if a pool is enabled or disabled, the standard verification options are provided, such as blocking until the pool is in the desired state, and polling at a given rate.

Syntax

To verify a pool state, use the following syntax:

```
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 PoolNameverify -poolstate
<-enabled | -disabled | -balancing>
```

Monitor thin pools

The `symcfg monitor` monitors the thin pools and displays the `-percent` usage.

Note

Monitoring thin pools is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

Syntax

To monitor usage of a thin pool, 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
```

Options

-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.

View thin pools

Description

When adding 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.

View thin device details

Examples

To list thin device details for array `086`, enter:

```
symcfg list -tdev -detail -sid 086
```

Sample output

```
. . .

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:

- A dash (-) displays in the Bound Pool Name column for all TDEV devices.
- A dash (-) displays 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 are displayed on subsequent lines.

View all thin device pools

Examples

To view all thin device pools for array 397, enter:

```
symcfg list -pool -all -sid 397
```

Sample output

```
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
```

View thin device allocations bound to a different pool

Description

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.

Examples

To show all allocations for pool `Mig_trg2` on array `432`, enter:

```
symcfg show -pool Mig_trg2 -thin -sid 432 -all -detail
```

Sample output

When the pool contains allocations from devices that are bound to a different pool:

```
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 10

Federated Tiered Storage

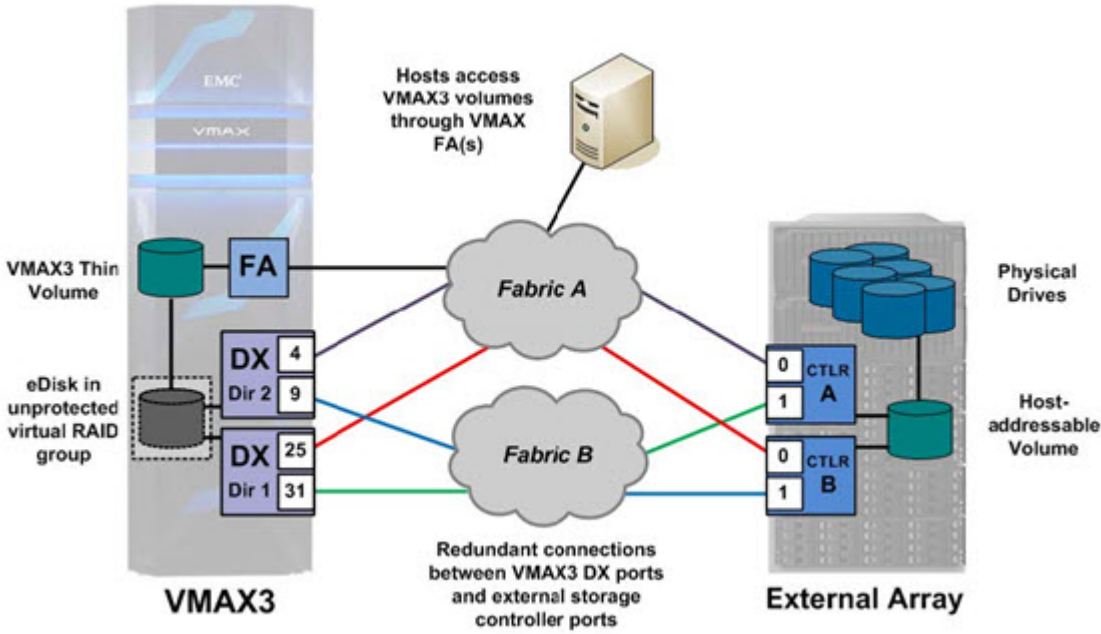
This chapter describes Federated Tiered Storage (FTS) concepts and explains how to configure and manage FTS using the SYMCLI.

- Federated Tiered Storage (FTS) and FAST.X 362
- eDisks 364
- Geometry of eDisk encapsulated devices (Enginuity 5876) 365
- Create external disk group 366
- Delete an external disk group 367
- Add an eDisk 368
- Show external disk information 369
- Remove an eDisk 370
- List external spindle state (HYPERMAX OS 5977 or higher) 371
- Verify eDisk state (HYPERMAX OS 5977 or higher) 371
- Start drain on eDisk (HYPERMAX OS 5977 or higher) 372
- Stop drain on eDisk (HYPERMAX OS 5977 or higher) 373
- Activate eDisk (HYPERMAX OS 5977 or higher) 374

Federated Tiered Storage (FTS) and FAST.X

FTS and FAST.X attach external storage to VMAX arrays and direct workload movement to these external arrays while having access to VMAX features such as local replication, remote replication, storage tiering, data management, and data migration. In addition, it simplifies multi-vendor or EMC storage array management. FTS and FAST.X use DX director emulation. The VMAX DX director is transparent to other director emulations and operating VMAX infrastructure, and allows the VMAX to act on the external logical units as if they were internal physical drives.

Figure 9 High-level overview of FAST.X environment



FTS and FAST.X array support is as follows:

- VMAX 10K, 20K, and 40K arrays, running Enginuity 5876 support FTS.
- VMAX 100K, 200K, and 400K arrays, running HYPERMAX OS 5977 or higher support FAST.X.

FAST.X overview for HYPERMAX OS 5977

FAST.X for VMAX arrays with HYPERMAX OS 5977 external provisioning includes:

- Data Domain eDisk encapsulation for EMC ProtectPoint — ProtectPoint allows for direct backup from VMAX arrays to Data Domain platforms. 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.

Note

The *Solutions Enabler TimeFinder Family CLI User Guide* provides details on using the SYMCLI to perform ProtectPoint operations.

- XtremIO, VNX, other VMAX3 eDisk connectivity — Integration with SLOs allows FAST management in these external arrays.

FTS overview for Enginuity 5876

Arrays running Enginuity 5876 support 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.*

Solutions Enabler features supported in FAST.X and FTS

Table 42 Solutions Enabler features supported in FAST.X and FTS

Feature	Enginuity 5876	HYPERMAX OS 5977
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.	FAST management integrated with SLOs.
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.	
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.	• Full support for externally provisioned device for ORS, and RP operations. • Support for encapsulated devices that are not geometry limited for only ORS and RP 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.	

Table 42 Solutions Enabler features supported in FAST.X and FTS (continued)

Feature	Enginuity 5876	HYPERMAX OS 5977
	• 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.	
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.	

eDisks

With a VMAX array attached to external storage, FTS and FAST.X virtualize an external array's SCSI logical units as VMAX disks called eDisks. eDisks have two modes of operation:

- Encapsulation — Preserves existing data on external arrays and accesses it through VMAX devices. These devices are called *encapsulated devices*.
- External Provisioning — Uses 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.

Restrictions

The following restrictions apply to eDisks:

- Must 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.

eDisk encapsulation

eDisk 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.
  For both modes, the array automatically creates the necessary VMAX devices. If the eDisk is larger than the maximum VMAX device size or the configured minimum auto meta size, the array creates multiple VMAX devices to account for the full size of the eDisk. These VMAX devices are concatenated into a single concatenated meta device and allow access to the complete volume of data available from the eDisk.

eDisk external provisioning

With a virtualized eDisk for external provisioning, VMAX devices can be created from the external disk group and used as storage.

For Enginuity 5876:
External provisioning is available for creating 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.

For HYPERMAX OS 5977:
Requires a new license (FAST.X) and Advanced Suite license pack.

Geometry of eDisk encapsulated devices (Enginuity 5876)

Enginuity 5876 builds VMAX devices based on the VMAX cylinder size (fifteen 64 K tracks), so the size of devices do not always match the raw capacity of the eDisk. If the size does not match, Enginuity sets a custom geometry on the encapsulated device. Solutions Enabler reports this custom geometry as user-defined geometry.

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.

eDisk Geometry limited devices restrictions (Enginuity 5876)

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:

- Only used as source devices for TimeFinder/Clone operations, no support for TimeFinder/Snap or TimeFinder/Mirror. 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.
- Only 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 expand, dissolve, convert, or use as the target of a VLUN migration.
- For thin devices, cannot unbind or rebind from their pool, and reclaim their space.
- For thin pools, created during encapsulation, cannot add data devices unless it results in encapsulating external storage.
- Cannot add thin pools to tiers.

eDisk encapsulation examples (Enginuity 5876)

This section provides examples of geometry-limited encapsulation and encapsulation that is not geometry limited.

Geometry-limited encapsulation
To 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)
To 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)
To virtualize an eDisk and encapsulate the data for a 22 GB external device, and 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.

Create external disk group

Description
eDisks require an external disk group (where eDisks are added).

Syntax
To create an external disk group, use the following syntax:

```
symconfigure create disk_group DskGrpName disk_location = <external>
```

Note

The disk group name serves as an additional identification mechanism and does not replace disk group numbers.

Example

To create a disk group for a storage pool named `hr_sg_pool`, enter:

```
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 allowed 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 maximum characters allowed in a disk group name is 32. 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.

Delete an external disk group

Syntax

To delete a disk group by disk group number or by disk group name, use the following syntax:

```
symconfigure delete disk_group <DskGrpNum | name:DskGrpName>;
```

Examples

To delete a disk group with the name `hr_disk_group`, enter:

```
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.

The following rules apply to physical disks and disk groups when using this feature:

- 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.

Add an eDisk

Description

When adding an eDisk to a VMAX array, it must be added to an external disk group. 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.

Syntax

For Enginuity 5876, add an eDisk to an external disk group, using 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 HYPERMAX OS 5977, add an eDisk to an external disk group, using the following syntax:

```
add external_disk wwn=<wwn> encapsulate_data=<YES | NO
<SRP=<SRPName>>>[dir=<Director_num>];
```

Options

encapsulate_data

Set this option to YES to encapsulate data on external LUNs. Set this option to NO to virtualize the eDisk for external provisioning. If set to NO, requires a valid SRP configured on the array.

Examples

For Enginuity 5876, to add an eDisk for external provisioning to disk group `exchange_disks1`, enter:

```
symconfigure commit -cmd "add external_disk wwn=60000970000194900306533030314341 to
disk_group=exchange_disks1, encapsulate_data=NO;"
```

For Enginuity 5876, to add an eDisk to disk group **`exchange_disks2`** and encapsulate the data using DP encapsulation, enter:

```
symconfigure commit -cmd "add external_disk wwn=60000970000194900306533030314345 to
disk_group=exchange_disks2, encapsulate_data=YES;"
```

For Enginuity 5876, to add an eDisk to disk group `exchange_disks2` and encapsulate the data using VP encapsulation (with a thin pool named `eng_pool`), enter:

```
symconfigure commit -cmd "add external_disk wwn=60000970000194900306533030314345 to
disk_group=exchange_disks2, encapsulate_data=YES, pool=eng_pool;"
```

For HYPERMAX OS 5977, to virtualize external LUN with WWN 6002198000002DDA8B into SRP "EXTERNAL_SRP", enter:

```
symconfigure –sid 087 commit -cmd "add external_disk wwn=6002198000002DDA8B
encapsulate_data=NO SRP=EXTERNAL_SRP;"
```

Rules and restrictions for adding eDisks

General restrictions

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.
- External LUN size must be between 42 cylinders and 16TB.

For HYPERMAX OS 5977:

- The specified SRP must exist, cannot have internal storage, and must have an external disk group. Only one external disk group is valid.
- The options `to disk_group` and `pool` are not valid.
- `Meta_member_size` or `member_size_equal` option cannot be specified.
- All external LUNs virtualized for external provisioning into a SRP should be from the same external array and should be the same size.
- A total of 2046 eDisks can be virtualized on the array. This includes both encapsulation and external provisioning.
- The Advanced Suite license pack and FAST.X license are required to perform this operation.

For Enginuity 5876:

- If using virtual provisioning, the name of an existing thin pool is required.
- To override the auto meta member size configured on the VMAX array, specify the meta member size (`meta_member_size =` *n* `[MB | GB | CYL]`).
- If `member_size_equal= NO` (specifies that devices do not have to be the same size), the meta tail will be smaller than the other devices in the meta.
- If the `encapsulate_data` option is set to NO, any data that is currently on the external device is deleted.

Show external disk information

Examples

To show external disk information for spindle ID `1E05`, enter:

```
symdisk show -spid 1E05 -sid 432
```

Sample output

```
Symmetrix ID                  : 000194900432
  Director                    : DX-2F
  Interface                   : N/A
  Target ID                   : N/A
  Spindle ID                  : 1E05
  External WWN                : 60000970000184700306533030314345
  External Array ID           : 000194900345
  External Device Name        : 00040
  Disk Group Number           : 512
  Disk Group Name             : DISK_GROUP_512
  Disk Location               : External
 ...
```

```
  Hypers (1):

  {
  #    Vol   Emulation        Dev   Type          Mir Mbr Sts Cap(MB)
  --- ----- ---------------- ----- ------------- --- --- --- --------
    1 N/A   FBA              00510 Ext-Data       1   1  RW
17261  }
```

Remove an eDisk

Description

For externally provisioning eDisks, the VMAX devices on the eDisk must be drained before it can be removed. See Start drain on eDisk (HYPERMAX OS 5977 or higher) on page 372on how to start draining an external disk.

For encapsulated eDisks, the devices on the eDisk must be unmapped and 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.

Syntax

To remove an eDisk, use the following syntax:

```
remove external_disk <wwn=<wwn> | spid=<SpindleID>>
[force_remove=<YES | NO>];
```

Options

force_remove

This option forces the removal of an external encapsulated disk when data is still in the VMAX cache. This can occur when the device has iVTOC and write pending tracks.

Examples

To remove an eDisk specifying the eDisks WWN, enter:

```
symconfigure commit -cmd "remove external_disk
wwn=60000970000184700306533030314345;"
```

To remove an eDisk using the specifying the spindle ID, enter:

```
symconfigure commit -cmd "remove external_disk spid=2256;"
```

Restrictions

- The following security privileges are required to execute the `remove external_disk` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- HYPERMAX OS 5977 requires Advanced Suite license pack and FAST.X license.
- The specified WWN or spindle ID must represent an external disk.
- The specified external disk must exist.
- The specified WWN or spindle ID must represent an external disk virtualized for external provisioning.
- The specified external disk must be in the *Drained* state.

List external spindle state (HYPERMAX OS 5977 or higher)

Syntax

To list the state of an external spindle, use the following syntax:

```
symdisk list -external –spindle –state [-spid <SymID>]
```

To list external spindle states for an array, use the following syntax:

```
symdisk list -external –spindle –state [-sid <#>]
```

Examples

To list external spindle states for array `084`, enter:

```
symdisk list -external -state -spindle -sid 084
```

Sample output

```
Symmetrix ID : 000197100084

          FLG                                          Drained
Spindle   T  Vendor   Product  Array ID     State     (%)
--------- --- ------- --------- ----------- -------- -------
8000      P  EMC     XtremIO  000012345678  Active    N/A
8001      P  EMC     XtremIO  000012345678  Draining  50
8002      P  EMC     XtremIO  000012345678  Drained   100

Legend:
 Flags:
  (T)ype: E - Encapsulated, P - External Provisioning
```

Output descriptions for `State` values:

- `N/A` — Value listed for array running Enginuity 5876.
- `Active` — At least one data device associated with the eDisk is enabled.
- `Draining` — At least one data device associated with the eDisk is draining.
- `Drained` — All data devices associated with the eDisk are disabled and not draining.
- `Drained (%)` — If the eDisk is fully allocated then the Drained % starts from 0. Otherwise it starts from a value depending upon the allocated tracks on the external disk. For example if the external disk is 100% allocated then the Drained % value starts from approximately 0%. If it is 70% allocated then the Drained % value starts at approximately 30%.
- `Disabled` — If draining is stopped using `stop drain`, the eDisk state becomes disabled and is not available for allocations.

Verify eDisk state (HYPERMAX OS 5977 or higher)

Syntax

To verify the state of an external disk, use the following syntax:

Note

The `ExternalWWN` argument for the `-spid` option must refer to an eDisk.

```
symdisk verify -external -sid <SymmID> [<-wwn=<ExternalWWN> | -spid=<SpindleID>>] [-i
<Interval>] [-c <Count>] < -draining | -drained | -active | disabled >
```

Examples

To verify the *Active* state for spindle `8000`, enter:

```
symdisk verify -external -sid 084 -spid 8000 -active

All disk(s) are in the 'Active' state.
```

To verify the *Draining* state for spindle `8000`, enter:

```
symdisk verify -external -sid 084 -spid 8000 -draining

None of the disk(s) are in the "Draining" state.
```

Start drain on eDisk (HYPERMAX OS 5977 or higher)

Description

The `start drain` command starts the background draining process. Use the `symdisk verfiy` or `symdisk list -state` commands to monitor the draining progress. (See Verify eDisk state (HYPERMAX OS 5977 or higher) on page 371 or List external spindle state (HYPERMAX OS 5977 or higher) on page 371). When draining is complete the external disk state is "Drained". If draining is stopped using the `stop drain` command, the external disk state changes from "Disabled" and the disk is not available for allocations. See Stop drain on eDisk (HYPERMAX OS 5977 or higher) on page 373 for `stop drain` command.

Syntax

To start the drain on an external disk, use the following syntax:

```
start drain on external_disk <wwn=<wwn> | spid=<SpindleID>>;
```

Examples

To start drain on external disk specifying the WWN, enter:

```
symconfigure -sid 087 commit -cmd "start drain on external_disk wwn=6002198000002DDA8B;"

Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
start drain on external_disk wwn=6002198000002DDA8B;
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session..............Done.
The configuration change session has successfully completed.
```

To start drain on external disk specifying the spindle ID, enter:

```
symconfigure –sid 087 commit -cmd “start drain on spid=23;”
```

```
Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
start drain on external_disk spid=23;
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session..............Done.
The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute the `start drain` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Requires Advanced Suite license pack and FAST.X license.
- The specified WWN or spindle ID must represent an external disk.
- The specified external disk must exist.
- The specified WWN or spindle ID must represent an external disk virtualized for external provisioning.
- The specified external disk cannot be in the *Draining* or *Drained* state.
- Sufficient free capacity must be available in the SRP, for all the allocated tracks produced from draining the external disk.

Stop drain on eDisk (HYPERMAX OS 5977 or higher)

Description

Stopping the drain process, moves the external disk state to *Disabled*.

Syntax

To stop the drain on an external disk, use the following syntax:

```
stop drain on external_disk <wwn=<wwn> | spid=<SpindleID>>;
```

Examples

To stop drain on external disk specifying the WWN, enter:

```
symconfigure –sid 087 commit -cmd “stop drain on external_disk
wwn=6002198000002DDA8B;”
```

```
Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
stop drain on external_disk wwn=6002198000002DDA8B;
    }
    Performing Access checks.................................Allowed.
      . . .
```

```
    Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

To stop drain on external disk specifying the spindle ID, enter:

```
symconfigure –sid 087 commit -cmd "stop drain on external_disk spid=23
```

```
Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
stop drain on external_disk spid=23;
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute the `start drain` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Requires Advanced Suite license pack and FAST.X license.
- The specified WWN or spindle ID must represent an external disk.
- The specified external disk must exist.
- The specified WWN or spindle ID must represent an external disk virtualized for external provisioning.
- The specified external disk cannot be in the *Drained* state.

Activate eDisk (HYPERMAX OS 5977 or higher)

Description

Activating an edisk is allowed when the external disk is in *DISABLED*, *DRAINED* or *DRAINING* state, and this action moves the edisk to *ACTIVE* state. Activating the edisk which is in *DRAINING* state stops the drain operation before activating.

Syntax

To activate an external disk, use the following syntax:

```
activate external_disk <wwn=<wwn> | spid=<SpindleID>>;
```

Examples

To activate an external disk specifying the WWN, enter:

```
symconfigure –sid 087 commit -cmd "activate external_disk
wwn=6002198000002DDA8B;"
```

```
Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
activate external_disk wwn=6002198000002DDA8B;
```

```
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session..............Done.
The configuration change session has successfully completed.
```

To activate an external disk specifying the spindle ID, enter:

```
symconfigure –sid 087 commit -cmd "activate external_disk spid=23;"
```

```
Execute a symconfigure operation for symmetrix '000197100087' (y/[n]) ? y
A Configuration Change operation is in progress. Please wait...
    Establishing a configuration change session..............Established.
    Processing symmetrix 000197100087
    {
activate external_disk spid=23;
    }
    Performing Access checks.................................Allowed.
      . . .
    Terminating the configuration change session..............Done.
The configuration change session has successfully completed.
```

Restrictions

- The following security privileges are required to execute the `start drain` command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Requires Advanced Suite license pack and FAST.X license.
- The specified WWN or spindle ID must represent an external disk.
- The specified external disk must exist.
- The specified WWN or spindle ID must represent an external disk virtualized for external provisioning.
- The specified external disk cannot be in the *Active* state.

CHAPTER 11

Device masking 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 378
- Create an auto-provisioning session 380
- Manage masking views 385
- Manage storage groups 387
- Manage port groups 390
- Manage initiator groups 393
- Display auto-provisioning group information 401

Auto-provisioning groups

Auto-provisioning groups creates groups of host initiators, front-end ports, and logical devices. These groups are associated to form a masking view, where 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 the `symaccess` command creates a group of devices, a group of director ports, a group of host initiators, and with one command, associates them into a *masking view*. Once a masking view exists, devices, ports, and initiators are easily added or removed from these 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 are 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 specific VMAX array OS versions are shown in Table 43 on page 378 and Table 44 on page 379.

Table 43 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 **Note** Using multiple child initiator groups in a cascaded initiator group with multiple initiators, allows a masking view to exceed the limit of 64 initiators.
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

Table 43 Provisioning limits for arrays running HYPERMAX OS 5977 or higher (continued)

Provisioning object	Configuration limitations
	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 44 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

Create a masking view

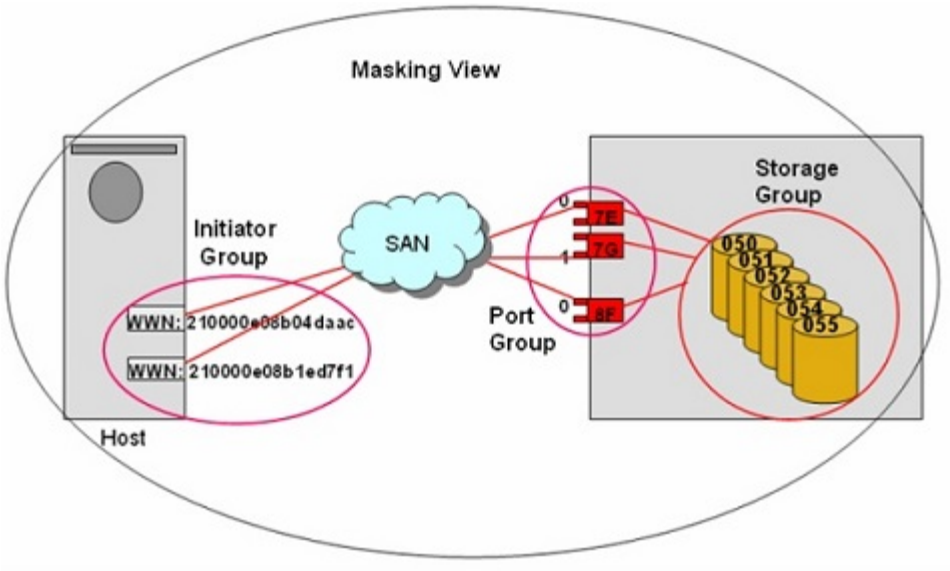
The steps for creating a masking view are:

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.

The devices are automatically masked and mapped when a masking view is created.

Figure 10 on page 380 shows a masking view.

Figure 10 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.

Auto-provisioning session rollback

Auto-provisioning operations (create masking views, or add devices or ACLX-enabled ports to existing views) continue to map devices to ports, but if the masking view update fails these devices are unmapped when the session is rolled back.
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) using the `-celerra`option. Storage groups containing devices tagged for RecoverPoint are masked and mapped using the `-rp` option.

HYPERMAX OS 5977 or higher

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, 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.

Enginuity 5876 or higher

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.

Create an auto-provisioning session

This section details the steps for creating an auto-provisioning session.

The steps are :

1. Discover the host HBAs.

2. List (identify) the host HBAs.
3. Create groups and views.

Discover host HBAs

Description
During the initial array setup, a search, from the controlling host, for each HBA connected to array devices is performed using the `symaccess discover` command. 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, identify their paths to the array.

Examples
To discover host HBAs: enter

```
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 is used with the `symaccess discover` command, to force a write of the discovered hostname/HBA name (or IP address) to the login history table and the initiator group.

2. Prints the initiator identifier (WWN/iSCSI) of the HBAs connected to the masked channel and the array.

List host HBAs

Description
For HYPERMAX OS 5977 and Enginuity 5874 or higher, both the `symaccess discover` and `symaccess list` commands do not use the ACLX device to determine the available connections. Therefore, connections available through all gatekeeper devices are listed regardless of the host visibility of the ACLX device. For the `symaccess list` command, the connections displayed include the paths detected through either the ACLX device or any gatekeeper device. If multiple gatekeepers, including the ACLX device, are available for detecting connections, only one entry is displayed for each unique connection to the array.

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.

Examples
To list host HBAs, enter:

```
symaccess list hba
```

Sample output
Shows only one unique connection through mutliple gatekeepers.

```
Identifier        Physical Device Path              Symmetrix ID
Dir:Port
----------------  --------------------------------  ------------
--------
10000000c99527b2  /dev/sdh                          000197100086  01E:
009
                  /dev/sdw                          000197300005  01E:
008
                  /dev/sdb                          000198700030  01E:
000
```

Create groups and views

The `symaccess` command is used to create the initiator groups, port groups, and storage groups that make up the masking view.

Create storage groups

Description
The `symaccess` and the `symsg` commands support storage group operations. Use the `symaccess` command for creating storage groups by specifying a range of devices, a list of devices, a device group, a storage group, or a device file.

Note

Host I/O Limits can be set 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 are consumed by a set of array devices over a set of director ports. Host I/O Limits for storage groups (5876 or 5977) on page 114 provides feature details and restrictions.

Syntax
To create a storage group, use the following syntax:

```
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...]]]]
```

Note

LUNs are assigned by the array when the masking view is created, not when devices are added to the storage group.

Examples
To create a storage group `SG_1` and add device range `050:055`, enter:

```
symaccess create -sid 458 -name SG_1 -type storage devs 050:055
```

Storage group and group name rules:

- Maximum group name length is 64 characters and are not case sensitive.
- Group Name must begin with an alphanumeric character, and embedded hyphens and underscore characters are valid.

- 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.

Create port groups

Description

A port can belong to more than one port group. However, for arrays running HYPERMAX OS 5977 or higher, different types of ports (physical FC ports, virtual FC ports and iSCSI virtual ports) cannot be mixed within a single port group.

Syntax

To create a port group, use the following syntax:

```
symaccess -sid SymmID-name GroupName -type port
create [-dirport Dir:Port[,Dir:Port...]]
```

Examples

To create a port group `PG_1` with three front-end ports, enter:

```
symaccess create -sid 458 -name PG_1 -type port -dirport 7E:0,7G:1,8F:
0
```

Create initiator groups

Description

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. Initiator groups are created using the HBA's WWN, iSCSI, a file containing WWNs or iSCSI names, or another initiator group name.

Syntax

To create an initiator group, use the following syntax:

```
symaccess -sid SymmID
create -name GroupName -type initiator
      [-consistent_lun]
      [-wwn wwn | -iscsi iscsi |
       -file InitiatorFilename | -ig InitiatorGroupName ]
```

Options

-consistent_lun

Use this 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, the host LUN number assigned to devices is the same for the ports on the HBA. If this option is not set, the system will choose the first available LUN on each individual port.

Examples

To create an initiator group, `IG_1`, enter:

```
symaccess create -sid 458 -name IG_1 -type initiator -file IG_1
```

The file `IG_1` contains:

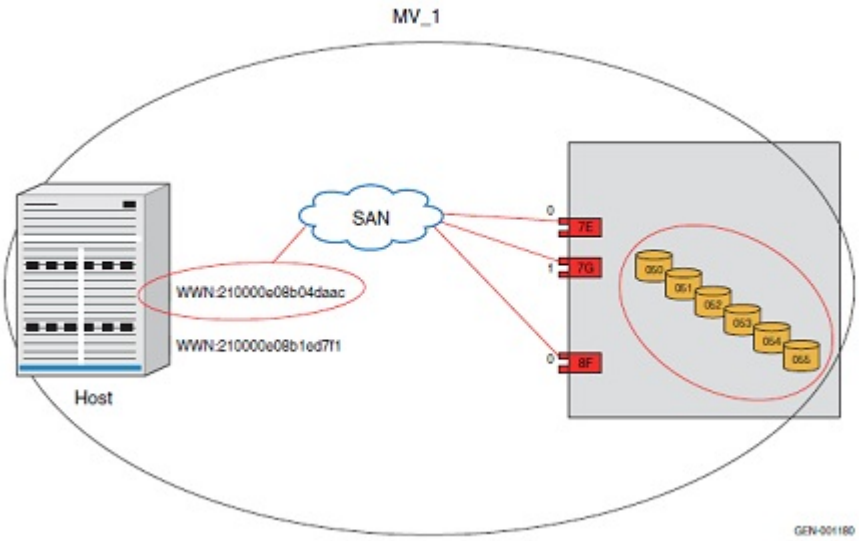
```
wwn:210000e08b04daac
```

Restrictions (HYPERMAX OS 5977 or higher)
For arrays running HYPERMAX OS 5977 or higher, different types of initiators (external Fibre Channel WWNs, internal guest Fibre Channel WWNs and iSCSI IQNs) cannot be mixed within a single Initiator Group. In addition, all child IG names added to a parent initiator group must contain the same Initiator type.

Create a masking view

Description
A masking view is a container of a storage group, a port group, and an initiator group, and makes the storage group visible to the host. Devices are masked and mapped automatically. The groups must contain some devices entries.

Figure 11 Masking view MV_1



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.

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.

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.

Syntax
To create a masking view, use the following syntax:

```
create view -sid SymmID -name ViewName -sg StorageGroupName
-pg PortGroupName -ig InitiatorGroupName
[ < [-reserve_id ResvID[,ResvID[,ResvID...]]]
[-lun Addr,Addr,Addr...] [-ckd] [-celerra] [-rp]
```

Options

-ckd
Masks CKD devices. By default, CKD devices are blocked from masking but are allowed if the devices are already mapped. Celerra devices can be masked without the flag if they are not mapped.

-celerra
Unnmaps Celerra devices.

-rp
Maps or masks devices tagged for RecoverPoint.

Examples
To create a masking view using `SG_1` `PG_1`, and `IG_1`, enter:

```
symaccess -sid 458 create view -name MV_1 -sg SG_1 -pg PG_1 -ig IG_1
```

After creating a masking view:
If additional storage is needed, devices added to the storage group are automatically masked and mapped in the masking view. This also applies to front-end ports and host initiators.

For managing other storage, create a second storage group and then create a masking view using 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.

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.

Manage masking views

This section explains how to perform the following management operations on masking views:

- Delete masking views
- Name groups and views
- Back up and restore views
- Copy groups and views

Delete masking views

Syntax

Note

When a masking view is deleted, all groups in the masking view remain intact. Any device reservations continue to be enforced when a masking view is deleted.

To delete a masking view, use the following syntax:

```
symaccess -sid SymmID delete view -name ViewName
```

Examples

For example, to delete masking view `MV_1`, on array **458**, enter:

```
symaccess -sid 458 delete view -name mv_1
```

Name groups and masking views

Syntax

To rename a storage group, port group, or an initiator group, using the following syntax:

```
symaccess -sid SymmIDrename -name GroupName
-type <storage | port | initiator> -new_name NewGroupName
```

To rename a masking view, using the following syntax:

```
symaccess -sid SymmIDrename view -name ViewName
-new_name NewViewName
```

If the new name already exists, an error returns.

Storage group and masking view naming rules:

- Maximum group name length is 64 characters and are not case sensitive.
- Group Name must begin with an alphanumeric character, and embedded hyphens and underscore characters are valid.
- 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`.
- Renaming of an initiator group is propagated to the higher group if the group is cascaded.

Back up and restore masking views

Description

The masking views, including storage groups, port groups, and initiator groups can be backed up to and restored from a file.

Syntax

To backup the masking views to a file, use the following syntax:

```
symaccess -sid SymmID -f BackupFilename [-noprompt]
   backup
   restore [-remove_ckd][-unused_sgs][-disassociate]
```

The `symaccess` command validates the consistency of the Auto-provisioning data before the `backup` or `restore` actions are performed.

Options

-noprompt
Eliminates the prompt for confirmation of the operation.

-remove_ckd
Skips all CKD devices within the backup, allowing the backup to be restored if the CKD devices are no longer mapped.

-disassociate
Disassociates the storage group from a FAST policy if the storage group contains invalid devices for FAST.

Examples

```
symaccess backup -sid 266 -f aclx_backup -nop
```

Example output with consistency errors:

```
Starting a backup operation.................
There are inconsistencies in the masking database. The operation
cannot be performed.
```

Example output with no consistency errors:

```
Starting a backup operation.................
The masking data on Symmetrix 000192600266 was backed up to file
aclx_backup.
```

Copy groups and masking views

Description

Groups and copy a storage, port, or initiator group, or a complete masking view can be copied from one array to another. When copying, any child view or cascaded initiator group are included in the copy action. The source and target arrays must be running Enginuity 5874 or higher.

Syntax

To groups or views from one array to another, use the following syntax:

```
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...]]]
```

Options

-reserve_id
Includes any device reservations.

-ckd
Specifies that the view contains CKD devices

-celerra
Specifies that the view contains Celerra devices (the devices will also be mapped).

-rp
Includes devices that have been tagged for RecoverPoint.

Examples

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
```

Manage storage groups

After creating a storage group, as explained in Create storage groups on page 382, the following actions can be performed:

- Add devices
- Remove devices
- Rename a storage group

- Delete storage groups
- List and show storage group information

Add devices to storage group

Description

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. Device reservations are still enforced when they are added 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.

Syntax

To add devices to an existing storage group, use the following syntax:

```
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...
```

Options

-ckd
Adds CKD devices to a storage group.

-celerra
Adds (and maps) Celerra devices to a storage group.

-rp
Adds devices tagged for RecoverPoint to a storage group.

LUN address designation

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.

AS400 and RecoverPoint-tagged devices

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.

Remove devices from storage group

Description

Deleting a storage group requires that all devices are removed. However a storage group cannot be completely emptied if it is associated with a masking view,

Syntax

To remove devices or child storage groups from a storage group, use the following syntax:

```
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
```

Examples

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
```

Rename a storage group

Syntax

To rename a storage group, port group, or an initiator group, using the following syntax:

```
symaccess -sid SymmIDrename -name GroupName -type <storage | port |
initiator> -new_name NewGroupName
```

Delete a storage group

Description

A storage group should be empty before it is deleted. It 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.

Syntax

To delete a storage group, use the following syntax:

```
symaccess -sid SymmIDdelete <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]
```

Options

-force
If the storage group still contains devices, forces the `delete` action.

-unmap
Unmaps 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 is not unmapped until the second masking view is deleted.

Examples
To delete storage group `SG_1` from array `458`, enter:

```
symaccess -sid 458 -name sg_1 -type storage delete
```

List and show storage group information

Description
Storage group information is requested from the array or from the backup file. Information for a device range or a list of devices can be specified.

Syntax
To view storage group information, use the following syntax:

```
symaccess -sid <SymmID>
[-offline] | -file <backup_filename>
     list -type storage
        [-devs <SymDevStart:SymDevEnd |
SymDevName | <SymDevName,SymDevName...>>]
        [-name GroupName] [-v | -detail]
     show GroupName -type storage
```

Manage port groups

Port groups contain director and port identification and belong to a masking view. Ports can be added and removed to the port group. Port group no longer associated with a masking view, can be deleted. In addition, CHAP authentication can be enabled and disabled on port groups. This section details the following port management operations:

- Add ports
- Remove ports
- Delete port groups
- Copy a port group from array to array
- List and show port group information
- Lock down a Fibre Channel ID

Note

The `symaccess` control operations `add, remove, backup`, and `restore` are modified to support the extended FA director port range.

Add ports to port group

Description
Ports and iSCSI targets are added to an existing port group by specifying the name and type of the group, and the director port or iSCSI target information.

Syntax
To add ports to a port group, using the following syntax:

```
symaccess -sid SymmID -name GroupName -type port[-ckd][-celerra] [-rp]
   add -dirport Dir:Port[,Dir:Port[,Dir:Port...]]
   add -iscsi_tgt iSCSI [,iSCSI...}
```

Options

-ckd

Specifies CKD devices.

-celerra
Specifies Celerra devices.

-rp
Specifies RecoverPoint devices.

Examples

To add port `4` of Fibre director `16D` to port group `PG_1` on array `245`, enter:

```
symaccess -sid 245 -name PG_1 -type port add -dirport 16D:4
```

Remove ports from port group

Syntax

Note

A port group cannot be emptied if it is associated with a masking view. To remove a port or an iSCSI target, use the following syntax:

```
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...]
```

Options

-unmap
Unmaps all ports in the port group associated with the view.

-force
Forces the port removal.

Example

To remove port `4` of Fibre director `16D` from port group `PG_1` on array `245`, enter:

```
symaccess -sid 245 -name PG_1 -type port remove -dirport 16D:4
```

HYPERMAX OS 5977 or higher

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) to the target array if it is running HYPERMAX OS 5977 or higher.

Delete port groups

Syntax

A port group cannot be deleted if it is associated with a masking view. To delete a port group, use the following syntax:

```
delete <view -name
ViewName> [-force] | -name GroupName
-type <storage | port | initiator > [-noprompt]
```

Options

-force
Forces the port group removal.

Examples

To delete port group `PG_1` on array `245`, enter:

```
symaccess -sid -name PG_1 -type port delete
```

Copy a port group from array to array

Syntax

To copy a port group from one array to another, use the following syntax:

```
symaccess -sid SymmID -target_sid SymmID copy -name GroupName -type
port
```

List and show port group information

Syntax

To display port group information, use the following syntax:

```
symaccess -sid <
SymmID  [-offline] | -file <
backup_filename>
     list -type port [-dirport Dir:Port]
           [-name GroupName] [-detail | -v]
     show GroupName -type port
```

Fibre Channel ID lockdown

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 networked systems in a shared (same director port) storage port configuration.

This feature sets Fibre Channel ID (FCID) of the WWN of the HBA to be secured. 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.

NOTICE

When an HBA logs in to a director port, the Fibre Channel ID accompanies it, indicating to 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.

Locking down a Fibre Channel ID

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:

Procedure

1. Find the WWN. If the device is visible, run the `symaccess list hba` command to find the device path of the HBA to protect.
2. Find the Fibre Channel ID value by referring to Find the FCID of a switch on page 442.

3. Run the `symaccess set lockdown` command on with the FCID of the Fibre Channel ID found in step 2.

   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
   ```

4. For the change to take effect, either reboot the host or pull the cable from the director and then replace the cable.

Manage initiator groups

This section describes how to manage initiator groups and includes the following tasks:

- Add initiators
- Remove initiators
- Delete initiators
- Initiator group flags
- Set HBA flags
- Replace a HBA
- Rename a HBA
- CHAP authentication

Add initiators to initiator group

Description

Initiators are added to an existing initiator group by specifying the initiator type (`-wwn` or `-iscsi`), the initiator group name, or by using an input file.

Syntax

To add initiators to an initiator group, use the following syntax:

```
symaccess -sid SymmID -name GroupName -type initiator
        -wwn wwn | -iscsi iscsi | -ig InitiatorGroupName |
         -f InitiatorFilename add
```

Add individual initiators to initiator group

Examples

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
```

Add initiators to initiator group using an input file

Examples

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
```

Note

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.

Remove initiators from initiator group

Syntax

To remove an initiator from an initiator group, use the following syntax:

```
symaccess -sid SymmID -name GroupName -type initiator -wwn
wwn | -iscsi iscsi | -ig InitiatorGroupName
        | -f InitiatorFilename [-login]
   remove
```

Options

-login

Removes the initiator from the array's login history table.

Examples

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
```

Deleting initiator groups

Syntax

To delete an initiator group, use the following syntax:

```
delete <view -name
ViewName> [-force] |
```

```
   -name
GroupName -type <storage | port | initiator > [-noprompt]
```

Examples

To delete initiator group `IG_1` on array `245`, enter:

```
symaccess -sid -name IG_1 -type initiator delete
```

Initiator group flags

Table 45 Initiator group flags

Initiator group	Flag
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]

Restrictions

- Modifications to the AS400 and Common_Serial_Number initiator group port flags are not allowed on arrays running HYPERMAX OS 5977 or higher.
- 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 clears the flag from the initiator group and all its initiators.

Set override flag for initiator group

Syntax

To set an override flag for an initiator group, use the following syntax:

```
symaccess -sid SymmID -name GroupName -type initiator
set ig_flags <on <Flag> <-enable |-disable> | off [Flag]>
```

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 individual initiator in the group.

Note

Modifications to the AS400 and Common_Serial_Number initiator group port flags are not allowed on arrays running HYPERMAX OS 5977 or higher.

Options

on
Turns on the specified initiator group port flag override and allows setting flag status of to *enabled* or *disabled*.

off
Turns off the specified initiator group port flag override.

enable
Sets the status of the initiator group port flag to *enabled*. The initiator group port flag override setting value must be `on` to set status.

disable
Sets the status of the initiator group port flag to *disabled*. The initiator group port flag override setting value must be `on` to set status.

Examples

To set the `OS2009 [OS2009]` flag for the initiator group `my_ig` on array `266`, enter:

```
symaccess -sid 266 -type init -name my_ig set ig_flags on OS2009 -
enable
```

To view the flag set in initiator group `my_ig` on array `266`, enter:

```
symaccess -sid 266 show my_ig -type init -detail
```

Sample output

```
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
```

Set HBA flags

Description

This feature allows specific host flags to be enabled and disabled on the director port. For arrays running Enginuity 5874 and higher, the HBA port flags are set on a per initiator basis, and the HBA must belong to an initiator group.

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.

Syntax

To set (or reset) the HBA flags, use the following syntax:

```
symaccess -sid SymmID -wwn wwn | -iscsi iscsi
set hba_flags <on <flag,flag,flag...> <-enable |-disable> |
       off [flag,flag,flag...]>
```

Options

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 as listed:

Table 46 Supported HBA flags

Supported HBA ports:	Supported initiator group ports:	Flag:
AS400	AS400	[AS4]
Avoid_Reset_Broadcast	Avoid_Reset_Broadcast	[ARB]
Common_Serial_Number	Common_Serial_Number	[C]
Disable_Q_Reset_on_UA	Disable_Q_Reset_on_UA	[D]
Environ_Set	Environ_Set	[E]
OpenVMS	OpenVMS	[OVMS]
SCSI_3	SCSI_3	[SC3]
SCSI_Support1	SCSI_Support1	[OS2007]
SPC2_Protocol_Version	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.

Examples

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 `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
```

Sample output

The `symaccess show -detail` output displays the flags that are turned on and off for each HBA initiator that has the feature enabled.

```
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 a HBA

If a host adapter fails, or needs replacement for any reason, assign the devices associated with the old adapter to a new adapter use the `replace` action with the following syntax:

```
symaccess replace -wwn wwn -new_wwn NewWWN [-noprompt]

symaccess replace -iscsi iscsi -new_iscsi NewiSCSI [-noprompt]
```

To swap HBAs:

Procedure

1. Run `symaccess list logins` to view the old WWN/iSCSI HBAs.
2. Swap the HBA boards according to the host instructions.
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.

Rename a HBA

Syntax

To rename the alias for a specified initiator within a group, use the following syntax:

```
symaccess -sid SymmID   rename -wwn wwn -alias alias
| -iscsi iscsi -alias alias
```

Using CHAP authentication

CHAP (Challenge Handshake Authentication Protocol) manages 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).

Enable CHAP on an iSCSI initiator

Syntax

To enable CHAP on an iSCSI initiator, use the following syntax:

```
symaccess -sid SymmID -iscsi iscsi enable chap
```

Enable CHAP on a director and port

Syntax

To enable CHAP on a director and port, use the following syntax:

```
symaccess -sid SymmID [-dirport Dir:Port]enable chap
```

Set the CHAP credential and secret

Syntax

To set the CHAP credential and secret, use the following syntax:

```
symaccess -sid SymmID -dirport Dir:Port| -iscsi iscsi
set chap -cred Credential -secret Secret
```

Disable CHAP on a specific director and port

Syntax

To disable CHAP on a specific director and port, use the following syntax:

```
symaccess -sid SymmID [-dirport Dir:Port] disable chap
```

Delete CHAP from a specific director and port

Syntax

To delete CHAP from a specific director and port, use the following syntax:

```
symaccess -sid SymmID [-dirport Dir:Port] delete chap
```

Display CHAP information

Examples

To display CHAP information for array `001`.

```
symaccess list chap -sid 001
```

Sample output

```
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
```

Verify the auto-provisioning database

Description

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.

Syntax

To verify the auto-provisioning database, use the following syntax:

```
symaccess -sid SymmID | -file BackupFileName  verify [-log]
```

Options

-log

Reports database inconsistencies in a log file

Examples

When database is consistent:

```
symaccess -sid 266 verify

Starting a verify operation.................
The auto provisioning database is consistent
```

Verification of a database backup file when database has 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
```

Display auto-provisioning group information

Syntax

To display auto-provisioning group information, use the following syntax:

```
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
```

Options

-offline

For use with 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, set the SYMCLI_OFFLINE environment variable to 1 to enable the offline mode for reporting.

-detail

Displays all auto-provisioning details. Without using the `-detail` option, any column without data does not display. If the `-detail` option is provided, the column without data displays a dash (`-`).

Modified reporting commands for HYPERMAX OS 5977

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 support the extended FA director port range:

- For `symaccess show -type port`, `show view`, and `symaccess show backupfile view` output, the `Director Identification` section supports 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 supports reporting for the extended port range.
- The `symaccess list chap` output supports 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 supports reporting for the extended port range.

Note

For arrays running HYPERMAX OS 5977, the `symaccess list no_assignments` output reports that "`All devices available on this director/port are assigned`" for ACLX enabled ports.

Display masking views

Description

The `symaccess show view` command lists the masking view and the associated initiator group, the port group, the storage group, and any child groups. This command also displays the time the group was modified and the time the associated masking view was modified.

Note

For arrays running Enginuity 5876 or lower, this command lists the array devices, director ports, device size, or the total capacity for the storage. For arrays running HYPERMAX OS 5977 or higher, if the masking view contains a cascaded initiator group, this command does not list this information and the `-detail` option must be used.

Options

-detail

Displays all masking view details, and columns without data displays a dash (`-`). Without using the `-detail` option, any column without data does not display.

Examples

To list all masking views on array `237`, enter:

```
symaccess -sid 237 list view
```

To list the details for masking view `view1_86`, enter:

```
symaccess -sid 0225 show view view1_86 -detail
```

Use the **`-detail`** option with the **`show`** command to display all the details of a view, including the child initiator groups.

Sample output

For all masking views on array `237`:

```
Symmetrix ID             : 000190300237
Masking View Name   Initiator Group Port Group Storage Group
------------------- ---------------- --------- --------------
View1                IG_1             PG_1     SG_1
View2                WinHost          PG
...
```

Detailed output for masking view `view1_86`:

```
Symmetrix ID                : 000196700255

Masking View Name           : view1_86
Last update time            : 04:36:39 AM on Mon Aug 31,2015
View last update time       : 04:36:39 AM on Mon Aug 31,2015

Initiator Group Name        : ig1_86
```

```
   Host Initiators
     {
       IG   : ig2_86
     }
Port Group Name             : pg1_86

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ----
-------------------------------------------------------
      FA-1D   027 500009735003fc1b
     }
Storage Group Name          : sg2_86

   Number of Storage Groups : 0
   Storage Group Names      : None

Masking View Name           : view1_86
Last update time            : 04:36:39 AM on Mon Aug 31,2015
View last update time       : 04:36:39 AM on Mon Aug 31,2015

Initiator Group Name        : ig2_86 *

   Host Initiators
     {
       WWN  : addddeb9db4bf88f
              [alias: addddeb9db4bf88f/addddeb9db4bf88f]
     }
Port Group Name             : pg1_86

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ----
-------------------------------------------------------
      FA-1D   027 500009735003fc1b
     }
Storage Group Name          : sg2_86

   Number of Storage Groups : 0
   Storage Group Names      : None

Sym                                          Host
Dev     Dir:Port  Physical Device Name     Lun   Attr  Cap(MB)
------  --------  ----------------------  ----  ----  -------
00060   01D:027   Not Visible                1          1796
                                                      -------
Total Capacity                                           1796

* Denotes a cascaded Initiator Group within the specified Masking View
```

View auto-provisioning group details

Syntax

To display storage group, port group, or initiator group details, using the `symaccess list` and `symaccess show` commands, use the following syntax:

```
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
```

Options

-type

Use this option to list groups of a specific type: (initiator, port, or storage).

-v

Lists details of a group and any children in the group.

-details

Shows detail of a masking view or any of the groups in the masking view.

Examples

To list details of a storage group or any child storage group, enter:

```
symaccess list -type storage -v
```

To show details of a masking view or any of the groups in the masking view, enter:

```
symaccess show view -detail
```

Sample output

For storage group:

```
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
```

For masking view:

```
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
```

View auto-provisioning device assignments

Syntax

To display the assignments for one or more devices, use the following syntax:

```
symaccess -sid
SymmID list assignments
   -devs
SymDevStart:
SymDevEnd |
SymDevName |
SymDevName,
SymDevName... [-v]
```

Examples

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
```

Sample output

```
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
```

Display device list with no auto-provisioning assignments

Syntax

To display a list of devices with no assignments, use the following syntax:

```
symaccess -sid SymmID list no_assignments
[-dirport DirNum:PortNum
```

Examples

To list the devices without assignments for array `120`, enter:

```
symaccess -sid 120 list no_assignments
```

Sample output

```
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
...
```

List initiator group devices

Syntax

To list all the devices masked to an initiator group, use the following syntax:

```
symaccess -sid SymmID list devinfo [-ig InitiatorGroupName]
```

When using the `-detail` option, the initiator group displays include the child initiator group device information.

View the HBA alias name

Example

To display the alias names for masking vew `host1082_view`, enter:

```
symaccess show host1082_view -sid 001 view
```

Sample output

Display supports up to 32 characters each for the alias node name and the alias port name.

```
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: api0190100000000C99DE136/
api0190100000000C99DE136]
       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 12

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.

- Device Masking overview 410
- VCMDB 410
- Host access process 410
- Device masking controls 411
- Supported device masking topologies 412
- HBA initiator support 412
- Initial device masking configuration 412
- Identify components for device masking 413
- Initialize and update the VCMDB database 415
- Enable iSCSI authentication 415
- Recommended actions for activating the VCMDB configuration 426
- Discovering host HBAs 427
- Add masked devices to HBA/director-port channel 428
- Remove masked devices 431
- Device masking database maintenance 433
- View the VCMDB 437
- Manage HBA initiators 440
- Fibre Channel-to-host interface management 442

Device Masking 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 are defined in the DMX-based device masking database (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 are also used to configure heterogeneous hosts with shared access to the same FA port, which is useful in an environment with different host types. However, Fibre Channel ID lockdown security is also used to protect an HBA from unauthorized 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 **`symmaccess`** command is used for device masking. For more information on using the `symaccess` command for VMAX arrays, refer to Device masking with Auto-Provisioning Groups on page 377.

VCMDB

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 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 control of 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 PowerPath is installed, then keep the VCMDB mapped.

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 after that grants access only to 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 first checks 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 are 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.

Access management for unsupported host platforms

Device masking can manage access for host platforms that are not supported by this release. If a host can log on to 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, manually set up a record in the database for these hosts.

Device masking controls

The `symmask` and `symmaskdb` commands specifically support the device masking control and monitor operations. These commands, 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` command locates paths to the device masking database (VCMDB) and assigns alias names (AWWN/AISCSI) to the NULL HBAs residing on the host where the command is run. The `rename` action is also used to generate aliases.

For detailed man page descriptions of these commands, refer to the *EMC Solutions Enabler CLI Command Reference*.

Device database locking

Description

When executing the `symmask` or `symmaskdb` command, 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 the host fails, or a Ctrl/C is performed in the while these command are running, the lock may 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, execution of a device masking command displays the following error message:

```
The operation failed because another process has an exclusive
lock on the local Symmetrix.
```

Syntax

To list external locks, use the following syntax:

```
symcfg -sid SymmID list -lock -lockn ALL
```

The command lists the external locks being held and the length of time it has been on.

Device database lock release

Syntax

To release this lock, use the following syntax:

NOTICE

Use the release action only if unlock operation was never run, and there are no other operations in progress to the database.

```
symcfg -sid SymmID -lockn LockNumber release
```

Supported device masking 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 types of 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 a native iSCSI environment, hosts are connected to an array through an Ethernet switch.

CHAP protocol

Native iSCSI support standards require the use of a security protocol. Enginuity version 5670 provides the Challenge Handshake Authentication Protocol (CHAP), which is enabled or disabled by the user. Refer to iSCSI software driver configuration on page 419 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). ASCII nicknames (AWWN and AISCSI) can be associated.

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 427 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 is added to the database so that access to the array is not lost.

Initial device masking configuration

Before using Solutions Enabler device masking, it is important to understand device masking for the environment, and scripts should be tested in a controlled environment.

During initial setup of the device masking environment, initialize the device masking VCMDB to clear devices of all data. See Initialize and update the VCMDB database on page 415.

Access control environment setup for device masking

If Symmetrix Access Control is being used to protect DMX devices, the host that is running the device masking commands must be configured in an access control group with an ACL (Access Control List) granting VLOGIX rights to ALL_DEVS.

Some controls require DIRCTRL or CFGSYM rights. Initialize, restore, and convert controls require CFGSYM and VLOGIX rights. Authentication and `symconnect` controls require DIRCTRL rights.

Note

Requires login to the control station as Administrator on a Windows system or as root on a UNIX system.

1. Identifying configuration components
2. Initializing and updating the database
3. Enabling authentication
4. Recommendations for activating the configuration

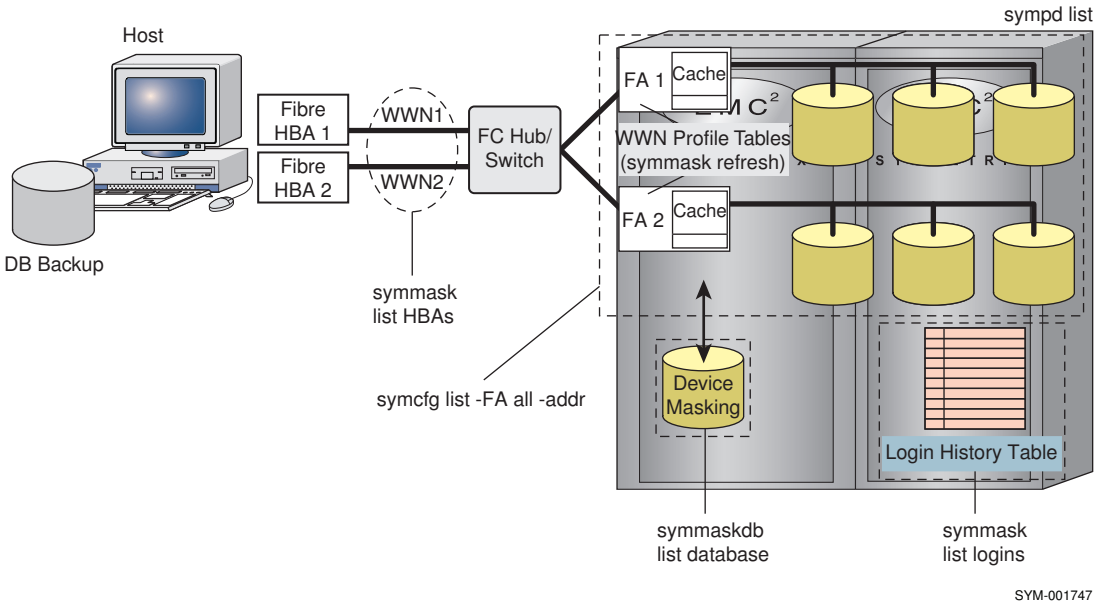
Identify components for device masking

Configuring device masking starts with Identifying components. This involves:

- Discovering local HBAs on a host that have a channel to a DMX array, run `symmask discover hba`.
- Generating an AWWN for any HBA that does not have an AWWN assigned.
- Updating the VCMDB with the new information.

SYMCLI device masking supports both World Wide Name (WWN) and native iSCSI (iSCSI) HBA connections.

Figure 12 Device masking components to identify



List physical device names of device-masking devices

Examples

To list the physical device names of all the device masking devices, enter:

```
sympd list -vcm
```

List HBAs and FA director ports

Examples

To list the HBAs on a host and the FA director port to which each HBA is connected, enter:

Note

Take note of the WWN of the HBA that is being configured and which FA port connects to that HBA.

```
symmask list hba
```

List host connections to DMX director ports

Examples

To List the DMX director port to which each HBA connects, enter:

```
symmask list logins
```

List devices mapped to FA director

Examples

To list devices mapped to the FA director 16A, enter:

```
symmask list logins
```

Initialize and update the VCMDB database

Description

After identifying components, initialize the database and create access records for HBAs to a directors. For more information on supported database types, refer to VCMDB database types on page 435.

Syntax

To initialize the database, use the following syntax:

```
symmaskdb -sid SymmID init -file BackupFilename[-vcmdb_type <3 | 4|
5>]
```

Examples

To add a range of devices (0030 through 0034) to the VCMDB, specifying the HBA's WWN and the FA director/port that the HBA connects to, enter:

```
symmask -sid 814 -wwn 20000000c920b484
add devs 0030:0034 -dir 16A -p 0
```

Enable iSCSI authentication

Before using the device masking SYMCLI commands with iSCSI, configure the 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 the RADIUS server, refer to Set RADIUS server configuration on page 425.

iSCSI CHAP authentication: Enginuity version 57xx

Beginning with Enginuity version 5771, connection security for iSCSI ports is set with the symconnect command. The symconnect functionality sets 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, 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 Set RADIUS server configuration on page 425.

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.

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 configuring a one-way authentication for each communication direction.

One-way CHAP authentication on page 416 shows how to set up the array as an authenticator, which covers one direction of communication. Two-way CHAP authentication on page 417 describes how to configure authentication from the opposite direction.

One-way CHAP authentication

Description

For one-way CHAP authentication, the `set chap` command automatically enables CHAP authentication, and defines the credential and the secret the host iSCSI initiator sends when challenged by the array.
Setting iSCSI authentication for an authenticator requires the array ID, the iSCSI name, the CHAP credential name (username), and the CHAP secret (password).

Syntax

To set the authentication for the iSCSI initiator in the CHAP authentication database with the name of `-iscsi iqn.2002-06.com.microsoft.host210`, use the following syntax:

```
symconnect -sid SymID -iscsi iqn.2002-06.com.microsoft.host210
set chap -cred MyCredentials -secret MySecret
```

Options

-cred
: Specifes the authentication data that the host should send in response to a challenge by the array. Windows hosts must specify a credential name string of 8 to 256 characters.

-secret
: Specifes the authentication data that the host should send in response to a challenge by the array. Windows host must specify a secret between 12 and 16 characters. UNIX hosts must specify a secret up to 32 ASCII characters, or 64 binary characters (binary values must use the prefix string **0x)** .

Examples

To disable CHAP authentication and re-enable it (turn it back on), enter:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210 disable
chap
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210 enable
chap
```

To disable authentication, enter:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210 delete
chap
```

Two-way CHAP authentication

Description

The `set chap` command automatically enables CHAP authentication for the array director and port. This defines the credential and the secret the array will send when challenged by the host HBA authenticator.

Syntax

To set up an array as a peer, enter:

```
symconnect -sid 6208 -dir # -p # set chap -cred MyPeerCredentials -
secret MyPeerSecret
```

Note

The host authenticator must have a user authentication database that contains matching authentication data for the authentication to be successful.

Options

-dir
Specifes the director on the array where the host HBA is connected.

-p
Specify the port on the array where the host HBA is connected.

-cred
Specifes the authentication data that the array should send in response to a challenge by the host. Windows hosts must specify a credential name string of 8 to 256 characters.

-secret
Specifes the authentication data that the array should send in response to a challenge by the host. Windows host must specify a secret between 12 and 16 characters. UNIX hosts must specify a secret up to 32 ASCII characters, or 64 binary characters (binary values must use the prefix string `0x`. This value is also set in the **iSCSI Initiator Properties** screen as described in Configuring iSCSI for two-way chap authentication on page 423.

Examples

To disable CHAP authentication and re-enable it (turn it back on), enter:

```
symconnect -sid 6208 -dir 2D -p 0 disable chap
symconnect -sid 6208 -dir 2D -p 0 enable chap
```

To disable authentication, enter:

```
symconnect -sid 6208 -dir 2D -p 0 delete chap
```

Display authentication information

Options

chap
Displays only the CHAP authentication database.

radius
Displays only RADIUS authentication database.

Examples

To display the authentication database (for example, any CHAP or RADIUS authentication database for array `20`), enter:

```
symconnect -sid 20 list
```

If CHAP credentials are defined in the authentication database for two iSCSI initiators, this makes it possible for the array to authenticate either of these iSCSI HBAs logging into its port. If credentials are defined for a director and port this allows for two-way authentication in the event that the array needs to submit authentication data to the host.

Initialize an authentication database

Description

The `symconnect init` command initializes and clears the data from authentication database. Prior to initializing an authentication database issue the `symconnect -sid Symid -list` command to list the CHAP or RADIUS databases and validate that the database exists.

NOTICE

This command is rarely used, as it clears data from database. Use with caution.

Options

-file
When initializing and clearing a database, specifies a file to write current authentication data to a backup location.

Syntax

To initialize CHAP or RADIUS authentication databases and write current authentication data to the backup files prior to initializing, use the following syntax:

```
symconnect -sid SymID init chap -file MyChapInitBackup
symconnect -sid SymID init radius -file MyRadiusInitBackup
```

Backup and restore authentication information

Backing up authentication information on a regular basis ensures that valid version of the authentication database can be restored if the current database becomes corrupted.

Backup authentication database

Syntax

To backup the CHAP authentication database, use the following syntax:

Note

Specify a new file with every backup.

```
symconnect -sid SymID -file MyChapBackup backup chap
```

Options

backup chap
Backs up the CHAP authentication database.

backup radius
Backs up the RADIUS authentication database.

Restore authentication database

Syntax

To restore the CHAP authentication database, use the following syntax:

```
symconnect -sid SymID -file MyChapBackup restore chap
```

Options

restore chap
Restores up the CHAP authentication database.

restore radius
Restores up the RADIUS authentication database.

List authentication database backup file

Syntax

To display the contents of a backup authentication file, use the the following syntax:

```
symconnect list -file MyChapBackup
```

iSCSI software driver configuration

The following are 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:

Note

Arrays must be installed and connected to the network before beginning the configuration.

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 allows the host computer to access the desired DMX devices:
   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
5. Configuration of the iSCSI driver is completed with CHAP authentication as described in Configuring iSCSI with CHAP authentication on page 420, or without

authentication as described in Configuring iSCSI without CHAP authentication on page 423.

Note

For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at http://elabnavigator.EMC.com

iSCSI software initiator location

The Microsoft iSCSI Software Initiator is available for download from the Microsoft download site.

Configuring iSCSI with CHAP authentication

Before you begin

After installing the iSCSI software initiator, these steps complete the iSCSI configuration with CHAP authentication. The iSCSI name of the host, and the iSCSI ID of the multi-protocol director are required for this configuration procedure.

These steps that require the use of the **iSCSI Initiator Properties** screen and the SYMCLI command window, as follows:

Collect information about the host computer and director

Procedure

1. Open the **iSCSI Initiator Properties** screen 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,

   ```
   symcfg -sid 6208 -dir 3a list -v
   ```

   The last two lines of the display contain the iSCSI name and IP address. as shown in this 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 syntax:

   ```
   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 different for each supported Enginuity version, as shown in the following syntax:

   - For Enginuity 5771 through 5773:

```
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 and higher (up to but not including Enginuity 5771):

```
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 415.

7. Refresh the VCMDB, as shown in the following syntax:

```
symmask -sid SymID 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 the 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** and start 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 the 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.
19. Perform disk administration such as, formatting, write signatures, and assigning drive letters.
20. Reboot the host system for configuration changes to take effect.

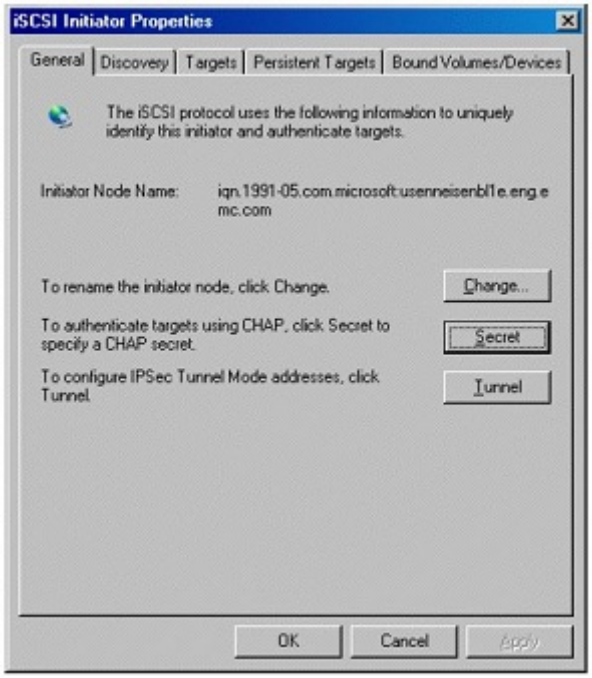
Configuring iSCSI for two-way chap authentication

Before you begin

After installing the iSCSI software initiator, these steps complete the iSCSI configuration for two-way CHAP authentication.

Procedure

1. Open the **iSCSI Initiator Properties** screen.
2. Click the **General** tab.
3. Click **Secret**.
4. Set the secret for two-way CHAP authentication to match the value set using **`symconnect`** as described in One-way CHAP authentication on page 416.



Configuring iSCSI without CHAP authentication

Before you begin

After installing the iSCSI software initiator, these steps complete the iSCSI configuration without CHAP authentication. The iSCSI name of the host, and the iSCSI ID of the multi-protocol director are required for this configuration procedure.

Note

This procedure contains steps that require the use of the **iSCSI Initiator Properties** screen and the SYMCLI command window, as follows.

Collect information about the host computer and DMX director

Procedure

1. Open the **iSCSI Initiator Properties** screen 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, as shown in this 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 syntax:

   ```
   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 syntax:

   ```
   symmask -sid SymID refresh
   ```

7. From the iSCSI Initiator window, select the **Target Portals** tab and click **Add**.
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** screen.
15. Perform any necessary disk administration, such as, formatting, write signatures, and assigning drive letters.
16. Reboot the host system for the configuration changes to take effect.

RADIUS server configuration

When using a RADIUS server to store authentication data, use the `symconnect` command to make the SYMAPI server recognize the RADIUS server. Once registered, the SYMAPI server looks to the RADIUS server for matching authentication data when attempting to authenticate a host's credential and secret.

Set RADIUS server configuration

Description

Setting the RADIUS server information at the director/port level, directs the array to the authentication information when a host HBA attempts to log in. the `symconnect set radius` command automatically enables RADIUS for the array's director and port.

Syntax

```
symconnect -sid SymID -dir # -p# set radius -server ServerName
-key MyServerPassword -ip ServerIPaddress -rank primary
```

Options

-rank
: 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**.

-port
: Specifies the RADIUS server port to be used if it is different to the server's default port.

-key
: Server password of up to 256 characters.

-i
: Specifies a server retry interval in seconds.

-c
: Specifies the number of times to retry.

Disable and enable RADIUS authentication for director/port

Examples

To disable RADIUS authentication on array `6208` for dir/port, enter:

```
symconnect -sid 6208 -dir 2D -p 0 disable radius -rank primary
```

To enable RADIUS authentication on array 6208 for dir/port, enter:

```
symconnect -sid 6208 -dir 2D -p 0 enable radius -rank primary
```

Enable or disable RADIUS authentication for specific iSCSI initiator

Description

After setting and enabling all RADIUS server information at the director and port level, enabling the server authentication for a given iSCSI HBA logging is allowed using either the `symconnect set` or `enable` command.

Examples

To enable RADIUS authentication on array 6208 for a specific iSCSI intitiator, enter:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210 enable
radius
```

To disable RADIUS authentication on array 6208 for a specific iSCSI intitiator, enter:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210 disable
radius
```

Delete RADIUS authentication

Examples

To delete the RADIUS authentication entry for a director/port, enter:

```
symconnect -sid 6208 -dir 2D -p 0 delete radius
```

To delete the RADIUS authentication entry for a specific iSCSI initiator, enter:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
```

Recommended actions for activating the VCMDB configuration

To ensure that updates to the VCMDB are active and visible to the host, the following steps are recommended:

1. Back up the device masking VCMDB to a file by running the `symmaskdb backup` command.

   Note

   Creating a backup is not allowed during the initial setup of the database.

2. Update the DMX array with the configuration changes by running the `symmask refresh` command. This refreshes the array director WWN/iSCSI-related profile tables in cache with the contents of the device masking VCMDB.

   Note

   Before running the `symmask refresh` command, or the `symmask -verify` command, verify that there are no HBAs accessing devices in the masked channel (applications running or user activity).

3. For changes to take effect, reboot all hosts that have had devices added or removed.

NOTICE

Run `symcfg discover` command. This scans the DMX devices and refreshes the SYMAPI configuration database.

4. When configuration of the database is complete, use the `sympd list` command to verify the correct devices are seen by the host.

Discovering host HBAs

Description

The `symmask discover` command is the primary mechanism used to identify for hosts, other than the control host, their VCMDB paths to the DMX array.

Options

-rename

Forces the discovered hostname/HBA name (or IP address) to be written to the login history table and the device masking VCMDB. This overwrites any existing AWWN/AISCSI record that were previously established.

Examples

To search the environment for DMX devices on each HBA, enter:

```
symmask discover hba
```

Note

It is assumed that the host running these commands has access to the DMX devices.

Results

When the `symmask discover` command 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 an alias is found, 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.
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 is used with the `symmask discover` command, the initiator identifier and its ASCII alias are written to the device masking VCMDB.
5. The `symmask discover` command sends information connection information back to the host.

HBA identifiers and alias names

Whether defined HBA alias names (AWWN/AISCSI) , or those assigned by the SYMAPI server during discovery are used for HBA identification, alias names can be used in the command line, replacing the complex numeric identifiers.

Alias names, which are stored in the DMX array's login history table, identify the HBAs connected to the network interface. Alias names are typically shorter in length and much more recognizable than cryptic WWNs/iSCSIs.

HBA identifiers and ASCII format

All alias names (auto-generated or user-defined) have two parts separated by a slash (/). An ASCII alias names generated by the `symmask discover` command consists of two parts: the name of the host and the name of the HBA, as follows.

- 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`, is `john4554b/10000000c920cf87`.
- For Native iSCSI configurations, the names are hostname/IP address.
  For more information on renaming Identifiers, refer to Manage HBA initiators on page 440.

Rename HBA identifiers

Description

The `symmask` command uses the AWWN or AISCSI to target specific HBAs for actions, such as adding or removing devices form the device mask. Renaming the AWWN or AISCSI for an HBA is allowed.

Syntax

```
symmask -sid SymmID -iscsi iscsi rename aiscsiNew
```

Examples

To change the name for HBA `20000000c920b484` to `Solar2b` on array `0128`, enter:

```
symmask -sid 0128 -wwn 20000000c920b484 rename Solar2b/b4
```

Note

To display the contents of the login history table and list the existing alias names, run the `symmask list logins` command.

Add masked devices to HBA/director-port channel

The DMX devices can be assigned to a specified masked channel (HBA to director port). 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 is found in the login history table.

Add masked devices

Syntax

To add devices to a specific HBA/director-port channel, use the following syntax:

```
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]
```

Options

-map
Maps the devices while adding them to the masking database.

Examples

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.

Add masked metadevices

When adding metadevices, add only the *SymDevname* of the device metahead.

Add masked reserved devices

Examples

If using devices that have been reserved, supply the device reservation ID. 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
```

Device reservation management on page 273 provides more information about device reservations.

Add masked device group devices

Syntax

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]
```

Options

If no device type is specified then only standard devices are added.

-std
Masks the standard devices only.

-bcv
Masks the BCV devices only.

-vdev
Masks the virtual devices only.

-tgt
Masks the target devices only.

Examples

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
```

Note

Device group names are not stored, only the device information is stored.

Add masked devices using a device file

Description

To add masked using a device file, first create the file, then include it in the `symmask` command.

Options

If source of target devices are not specified the only source devices are added.

src
Masks only the source devices listed in the device file.

tgt
Masks only the target devices listed in the device file.

Examples

To create a device file named `mask2`, enter the source devices first, a space, then the target device, and use the pound sign (#) for comment lines, as follows:

```
#Device list mask2 for masking
0014 009A
0151 014C
```

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
```

Add devices with dynamic LUN addressing

Syntax

To add a device or devices to a specified HBA/director-port channel, specifying the LUN address, use the following syntax:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi |-aiscsi
aiscsiadd devs SymDevStart:SymDevEnd|SymDevName|SymDevName,,,...\
-lun Addr | Addr, Addr, Addr...
-dir # -p # [-noprompt][-dynamic_lun] [-remove_unmapped] [-map]
```

Options

-lun
Specifies starting LUN addresses. Single starting LUN or multiple LUNs to match the given ranges can be used.

-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

Removes 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

Maps devices while adding them to the masking database. This option takes the next available LUN, which may or may not match the masking LUN. This option is only supported on Enginuity 5773.

Examples

To specify a single starting LUN or multiple LUNs to match the given ranges, enter:

```
symmask add devs 15,18,20 -lun 0 -wwn 20000000c920b484 -dir 14C -p 1 -sid 12
```

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
```

To assign the LUN addresses, when adding devices from a device file or list of devices, enter:

```
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
```

Remove masked devices

Description

Selected devices or an entire set of devices associated with an masked channel can be removed . Removing selected devices requires syntax similar to adding masked devices, and the same refresh, backup, and discover steps after completing the remove operation.

Syntax

To remove a device or devices from a masked channel, use the following syntax:

```
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]
```

Options

-unmap

Unmaps 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.

-force

Removes a range of specified device names that might span device names not part of the existing noncontiguous masked channel, and may be needed when working with metadevices.

Examples

To remove devices `0031` and `0033` from the 0030-to-0034 range of devices, enter:

```
symmask -sid 814 -wwn 20000000c920b484 remove devs 0031,0033 -dir 16A -p 0
```

After removing devices, activate the masked configuration as follows:

1. Run the `symmask refresh` command.
2. Run the `symmaskdb backup` command.
3. Reboot the host.
4. Run the `symcfg discover` command refresh the SYMAPI database.

Remove masked metadevice members

Syntax

To remove metadevice members from the device masking VCMDB, but keep the metaheads in place, use the following syntax:

```
symmaskdb -sid SymmID -meta_member remove -force
```

Options

-force
Forces the remove action and may be needed when removing metamembers.

Remove masked devices from device group

Syntax

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 aiscsiremove
-g DgName [-std] [-bcv] [-vdev] [-force] -dir # -p # [-noprompt]
```

Options

If no device type option is specified, only standard devices are removed.

-std
Removes the standard devices only.

-bcv
Removes the BCV devices only.

-vdev
Removes the virtual devices only.

Examples

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
```

Note

Device group names are not stored, only the device information is stored.

Remove devices using a device file

Options

If source or target devices are not specified, only source devices are removed.

-src
Removes the standard devices only.

-tgt
Removes the BCV devices only.

Examples

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
```

Device masking database maintenance

Each time a SYMCLI command is issued, that affects device masking, the VCMDB changes. These database changes may include:

- Adding, updating and deleting records for devices
- Renamed WWN /iSCSI alias
- One HBA swapped for another
- Clearing an HBA

These changes require database maintenance to ensure that records are not lost. Some of these maintenance tasks include:

- Syncing masking data
- Creating a backup file
- Restoring a backup file
- Blocking direct writes to the database

View masking database

Examples

To view the VCMDB on array `0128`:

```
symmaskdb -sid 0128 list database
```

Initialize and clear masking database

Description

For the initial setup of any device masking environment, a device reserved (VCM state enabled) in the DMX array is initialized and formatted for use as the device masking database.

NOTICE

Initializing an existing database clears the disk device of any current data in the process of formatting the database. This command is rarely used, as it clears data from database. Use with caution.

Syntax

Note

Specify a backup filename, as this command writes the data from this device to a backup file on your host before it clears the current data.

To initialize and clear the database device, use the following syntax:

```
symmaskdb -sid SymmID init -file BackupFilename[-vcmdb_type <3 | 4|
5>]
```

If you do not include the **-vcmdb_type** option, the default database type depends on the size of the VCMDB device.

Options

-vcmdb_type

Defines the number of devices per database record and the number of records that can be masked. This can be used to address backwards capatiblity issues:

- 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).

Note

There is a VCMDB Type 6, but it cannot be specified with SYMCLI.

Examples

To initialize the database and create a backup file `BackupDevMask1` on array `0128`, enter:

```
symmaskdb -sid 0128 init -file BackupDevMask1
```

To initialize the database as VCMDB type `4` and create a backup file `MyInitBackup` on array `0814`, enter:

```
symmaskdb -sid 814 init -file MyInitBackup -vcmdb_type 4
```

Sync masking data

Examples

For example, to sync masking data on array **0128**, enter:

```
symcfg sync -sid 0128 -masking
```

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.

Create VCMDB backup file

Description

A backup file containing the current contents of the device masking VCMDB can be created, and is useful when temporarily changing the access rights or device masking assignments to various HBAs. Restoring the backup file returns the device masking environment back to the original masked environment.

Syntax

Note

Reusing a backup file name is not allowed. Always create a new file with each backup.

To create a backup database file, use the following syntax:

```
symmaskdb -sid SymmID backup -file BackupFilename
```

The VCMDB backup files vary in length, depending on the contents.

Examples

To create backup file `BackupDevMask` on array `0128`, enter:

```
symmaskdb -sid 0128 backup -file BackupDevMask
```

VCMDB database types

Device masking database types define the number of devices per database record and the number of records you can mask.

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 is created for a 48-cylinder device. This default behavior is also true for the VCMDB device that currently holds a database (that is, a Type 4 database on a 96-cylinder device is 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. Using the **`set no_direct_io`** option blocks direct I/O writes to a Type 3 database .

Convert a VCMDB type

Description

A Type 3 database can be converted 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.

If a lower type database is converted 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 no longer has access to the database until this host is upgraded to version 6.0 or higher.

Example
To convert the VCMDB on array `814` to a Type `5` database, enter:

```
symmaskdb -sid 814 convert -vcmdb_type 5 -file MyCvrtBU
```

Restore a backup VCMDB

Description
The database is restored as is; Type 3 restores as a Type 3, and Type 4 restores as a Type 4. Use the `convert` or `set` options to alter the resulting environments.

Examples
To restore the database from the backup file `MyCvrtBU` stored on the host, enter:

```
symmaskdb -sid 0128 restore -file MyCvrtBU
```

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 or higher.

Restore a backup and converting its type

Example

Note

A Type 3 database can be converted 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.

To restore from a backup file and convert its database type in a single command, and specify type 5. enter:

```
symmaskdb -sid 0128 restore -file MyCvrtBU -vcm_type 5
```

Block direct writes to the VCMDB

Syntax
To block direct writes to a Type 3 database, use the following syntax:

```
symmaskdb -sid SymmID set -no_direct_io
```

Note

Type 4 and 5 databases initialize with direct I/O writes blocked by default.

View the VCMDB

When the database is initialized and records created, information about the database can be viewed. The login history details can also be viewed.

View contents of VCMDB

Syntax

To view the entire contents of the device masking VCMDB, or by director/port, WWN (or alias), or iSCSI name (or alias), use the following syntax:

```
symmaskdb -sid SymmID | -file Filename [-v]
[-dir all [-p all] | -dir
#
 [-p <
#
|all>]]
[-wwn wwn | -awwn awwn |
-iscsi iscsi | -aiscsi aiscsi]
list database
```

Example

To view the entire device masking VCMDB on array `6196`, enter:

```
symmaskdb -sid 6196 list database
```

View VCMDB records for a director

Example

To view the database for records concerning director `2b`, port `4` on array `6196`, enter:

```
symmaskdb -sid 6196 list database -dir 2b -p 4
```

View masked assignment to specific HBA

Syntax

To view the masked assignment of devices to a specific HBA, use the following syntax:

```
symmaskdb -sid SymmID list devs [-wwn wwn | -awwn awwn|-iscsi iscsi |
-aiscsi aiscsi
```

Note

he initiator is optional.

Example

To view the devices on array `343` without including a specific initiator, enter:

```
symmaskdb -sid 343 list devs
```

View device capacity

Description

When a hostname is specified, SYMAPI looks for records with the AWWN value of the host `(wwn)` and pulls 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.

Syntax
To view the capacity of devices assigned to a particular host, use following syntax:

Note

This command requires that the first part of the HBA alias be the host name.

```
symmaskdb -sid SymmID list capacity -host HostName
```

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 to specify a backup filename.

Example
To view the capacity of host `api145` on array `6196`, enter:

```
symmaskdb -sid 6196 list capacity -host api145
```

View HBA assignments

Options

no_assignment
Lists devices that are mapped, but not yet assigned in the device masking VCMDB

Example
To view which HBAs have been assigned to specific devices, enter:

```
symmaskdb -sid 6196 list assignment -devs 0040:0043
```

View the login history table

Description
Use the `symmask list logins` command 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. Options can restrict this data to a specific director and/or port number.

Options

-v
Displays the last active login information.

Example
To return login information for array `6196`, enter:

```
symmask -sid 6196 list logins
```

Sample output

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
```

Output fields

- `Identifier` — lists which HBA is communicating with the array.
- `User-generated node name` — lists the AWWN or AISCSI alias associated with the node.
- `User-generated port name` — lists the AWWN or AISCSI alias associated with the port.
- `On Fabric` — indicates if the HBA is connected to a fabric and if it is logged in to the DMX system.
- `Logged In` — indicats if the HBA is connected to a fabric and if it is logged in to the DMX system.

Refresh the database

Description

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 pushes all records within the database out to the local directors. A login by an HBA updates the record only for the HBA.

The output for the `symmask refresh` command is similar to the `symmaskdb list`. The differences are:

- Lists 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.

Options

-verify

Views what is currently stored on the local directors to see what (if any) affect a refresh will have.

Example

To view what is currently stored on the local directors for array `0237` and see the affect of a refresh, enter:

```
symmask refresh -verify -sid 0237
```

Manage 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. Rarely, an HBA has four ports. HBA identification in device masking always refers to a port WWN.

List AWWN logins

Description
When issuing 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.

Note

The AWWN can be renamed to a shorter name.

Example
To list AWWNs in the login history table of array `814`, enter:

```
symmask -sid 814 list logins
```

Rename AWWNs

Description
Use the **`symmask rename`** command to assign an AWWN. Assign an AWWN name at any time, even before SYMAPI generates an AWWN.

In `symmask` commands HBAs are identified using an iSCSI name. iSCSI is a SCSI-over-IP protocol that uses 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, the `symmask discover hba`command a generates a two-part alias name.

Examples
To assign the two-part name `Solaris3A/b4` as the AWWN for WWN `20000000c920b484`, enter:

```
symmask -sid 814 -wwn 20000000c920b484 rename Solaris3A/b4
```

To replace all user-defined AWWNs with system-generated AWWNs, and overwrite all existing AWWN entries in the VCMDB, enter:

```
symmask discover hba -rename
```

Replacing an HBA

If a host adapter fails or needs replacement for any reason, replace the adapter and assign its set of devices to a new adapter using the following syntax:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi|-aiscsi aiscsi
replace wwnNew | iscsiNew
```

To swap HBAs:

Procedure

1. Run the `symmask list logins` command to view the old WWN/iSCSI HBAs.
2. Swap HBA boards.
3. Run the `symmask list hba` or `discover` command to view the new initiator (for example WWN).
4. Run the `symmask replace` command to substitute a new WWN for all occurrences in the database of the old WWN.

   To replace old WWN `20000000c920b484` with new WWN `20000000c920b393`, enter:

   ```
   symmask -sid 814 -wwn 20000000c920b484 replace 20000000c920b393
   ```

5. Run the `symmask discover -rename` or `symmask rename` command to establish the new AWWN and assign an AWWN to the new HBA in both the VCMDB and the login history table.
6. Run the `symmask refresh` command to update the director profile tables (in cache) from the database.

Delete HBA associations

Syntax

To delete (in the database) the set of devices associated to a host adapter, use the following syntax:

```
symmask -sid SymmID delete -wwn wwn|-awwn awwn|-iscsi iscsi
|-aiscsi aiscsi[-login]
```

Options

-dir
Restricts the action to only devices on the specified director.

-p
Restricts the action to only devices on the specified port.

Delete HBA associations

Description

On arrays running Enginuity 5771 through 5773, use the `-login` option to also delete the entry from the login history table.

Syntax

To delete (in the database) the set of devices associated to a host adapter, use the following syntax:

```
symmask -sid SymmID delete -wwn wwn|-awwn awwn|-iscsi iscsi
|-aiscsi aiscsi[-login]
```

Options

-login

Deletes HBA association from the login history table

After the `delete` action, use the `symmask refresh` command to update the array with the configuration change. This refreshes the WWN/iSCSI-related profile tables in cache with the contents of the device masking VCMDB.

NOTICE

Before running the `symmask refresh` or the `symmask refresh -verify`command, make sure there are no HBAs accessing devices in the masked channel (applications running or user activity).

Fibre Channel-to-host interface management

Use the device masking commands to adjust the protocol characteristics of the Fibre Channel-to-host interface to be compatible with the host platform-specific requirements. A record for the host adapter port assignment must already exist in the VCMDB for these channel attributes to be set.

NOTICE

Advanced understanding of the HBA interfaces is required to adjust these characteristics. Improper settings can disable the use the host with the array.

The `symmask set` command adjusts the following attributes on a host adapter port basis, or all ports using `-dir ALL` / `-p ALL` options:

- Fibre Channel ID (FCID) lock down
- Device LUN visibility
- LUN base/offset skip
- Heterogeneous host configuration

Find the FCID of a switch

To find the Fibre Channel ID on Connectrix or Brocade switches:

- Connectrix switch:
  1. Through the hardware view, click the board.
  2. Click the port of the switch whose Fibre Channel ID you want to find.
  3. Right-click to display the port properties window that includes the FCID value.
- Brocade switch:
  1. Telnet to the switch and run `nsShow`.

2. Look for the PID value of the WWN of the HBA you want to protect, which is the Fibre Channel ID value.

Fibre Channel ID lockdown

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 networked systems in a shared (same director port) storage port configuration.

This feature sets the Fibre Channel ID (FCID) of the WWN of the HBA to protect it. 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.

NOTICE

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.

Locking down FCID

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:

Procedure

1. Find the WWN. If the device for the device masking VCMDB is visible, run the `symmask list hba` command 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 the `symmask list logins -pdev` command , specifying the device path found in step 1, to find the Fibre Channel ID of the WWN of the HBA to lock down.
   - Find the Fibre Channel ID value on the switch, refer to Find the FCID of a switch on page 442.

3. Run the `symmask set lockdown`command with the FCID of the Fibre Channel ID found in step 2.

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
```

4. Run the `symmaskdb list database`command 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 rebooting, run the `symcfg discover` command to refresh the SYMAPI database.

FCID lockdown 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.

If there is no path from the HBA to Fibre Channel ID to be locked down, The `symmask list logins` command cannot be used to find the FCID value; it must obtain the FCID value from the switch.

FCID format

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:

*22***04**13
Italicized text is the domain. Bold text is the port. The domain is 2 and the port is 04

Note

For Connectrix, the port is offset by 4.

Brocade 2000 series DS-16B:

*02*1**3**00
Italicized text is the domain. Bold text is the port. The domain is 02 and the port is 3.

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, such as Linux, do not detect the remaining LUNs and fail to discover noncontiguous devices.

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 a LINUX 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 Set LUN base/offset skip adjustment on page 445.

Set device LUN visibility

To allow a host to detect 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.

Example

To turn on visibility for FA director `16A`, port `0`, when working with a host HBA whose WWN is `20000000c920b484`, enter:

```
symmask -sid 814 set visibility on -dir 16A -p 0 -wwn 20000000c920b484
```

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 an issue when needing 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 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).

Set LUN base/offset skip adjustment

Syntax

To set LUN base and offset values for a skip hole within an HBA to director channel, use the following syntax:

```
symmask -sid SymmID set lunoffset on offset base -awwn awwn -dir # -p #
```

Example

To make LUNs (devices) `005` through `008` (with director `16A`/port `0`) available to host `HPB03/1`, with a LUN base address of `000` and an offset of `5` (to skip over 000-004), enter:

```
symmask -sid 018 set lunoffset on 5 0 -awwn HPB03/1 -dir 16A -p 0
```

Multiple hosts and broken sequences

If masked devices for an HBA-to-FA connection had LUN addresses 0000-0003 and 0007-0009, specify a LUN base address of 0004 and an offset of 0003 (to renumber LUNs 0007-0009 as LUNs 0004-0006). Only one gap per HBA-to-FA connection can be recorded.

If multiple hosts that cannot discover devices with noncontiguous LUN addresses are present, issue the `symmask set lunoffset` command for each host.

As shown in Host platforms and interface configuration flags on page 448 for Scenario 1, LUN devices 000 through 006 are assigned to Host A and 007 through 00A assigned to Host B. Host B needs to see LUN 000 first, so use `set lunoffset on` with a base value of 000 and an offset of 7. Host A has no issue since there is no hole in its assigned device sequence and it starts with 000.

For Scenario 2, LUNs 000 through 002 and 007 through 008 are assigned to Host A. Host B could have LUNs 003 through 006. Host A's 000-002 is not a issue, 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.

Table 47 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

HBA port flags

HBA port flags can be set on a per initiator basis or by host name. This feature enables and disables specific host flags 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 or higher.

Setting HBA port flags replaces setting the heterogeneous host configuration flags (Set heterogeneous host configuration on page 448). The heterogeneous host configuration types listed in Table 49 on page 448 continue to be valid, but will not be expanded. To switch to setting HBA port flags, disable the heterogeneous host configuration for a given director/port/HBA and reset all flags.

Set HBA port flags

Syntax

To set (or reset) the HBA flags, use the following syntax:

```
symmask -sid SymmID -wwn wwn | -awwn awwn |
        -iscsi iscsi | -aiscsi aiscsi | -host HostNameset hba_flags
           <on <<flag>,<flag>...> <-enable | -disable> |
            off [<flag>,<flag>...] >
           -dir <#|all> -p <#|all>
```

Options

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]:

Table 48 HBA port flags

Common_Serial_Number [C]	AS400 [AS4]OpenVMS [OVMS]
Disable_Q_Reset_on_UA [D]	SCSI_3 [SC3]
Environ_Set [E]	Sunapee [SCL]
Siemens [S]	Sequent [SEQ]
Volume_Set_Addressing [V]	SPC2_Protocol_Version [SPC2]
Avoid_Reset_Broadcast [ARB]	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.

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
```

Heterogeneous host configuration

Note

Setting the heterogeneous host configuration is superseded by setting the HBA port flags (Set HBA port flags on page 446). The heterogeneous host configuration types listed in Table 49 on page 448 are valid, but will not be expanded. To switch to setting HBA port flags, disable the heterogeneous host configuration on the array and reset all flags.

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 interface protocol.

Turning on heterogeneous host, turns on the bits listed in Table 49 on page 448, (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, so the devices they see are different from those seen by any other host on the director.

Set heterogeneous host configuration

Syntax

To set heterogeneous host configuration flags and optimize the host-to-director interface, use the following syntax:

```
symmask -sid SymmID -wwn wwn -dir #
 -p # set heterogeneous on HostConfigFlag
```

Possible *HostConfigFlag* values are listed in the last column of Table 49 on page 448.

Host platforms and interface configuration flags

Table 49 Host platforms and interface configuration flags

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	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

Table 49 Host platforms and interface configuration flags (continued)

Host platform	Requirements	Bit [a]	Host configuration flag (*HostConfigFlag*)
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
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

Table 49 Host platforms and interface configuration flags (continued)

Host platform	Requirements	Bit [a]	Host configuration flag (*HostConfigFlag*)
VMware (ESX 2.5 or earlier only)		C, SC3*	VMWARE

a. 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 13

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.

- Quality of Service overview 452
- Copy pace and priority 452
- Host I/O priority (Enginuity 5876 or lower) 455
- Set bandwidth for Open Replicator (Enginuity 5876 or lower) 457
- Set and reset LRU parameters 457
- RDF director workload distribution 458
- Dynamic cache partitions (Enginuity 5876 or lower) 460

Quality of Service overview

The QoS (Quality of Service) SYMCLI feature adjusts the data transfer (copy) pace on specified devices, or devices in a device group or storage group, of certain operations.

QoS increases the response time for specific copy operations on selected devices or groups, and 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.)

Copy pace and priority

The `symqos` command sets 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.

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 this case, applying QoS on the source applies to all copy sessions. Setting on the target applies to that particular session.

A different clone copy priority can be set 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.

Set copy pace and priority

Description

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.

Syntax

To set the copy pace, use the following syntax:

```
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
```

Options

If a device group is not specified, a storage group or devices must be specified.

-dg
The copy operation is on a device group devices.

-sg
The copy operation is on storage group devices.

-bcv
The copy operation works only on BCV group members.

-nobcv
The copy operation works only on group members that are not BCVs.

-devs
The copy operation is on a range of devices.

BCVs
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

Enginuity 5875 or 5876 is required to set the pace/priority for VLUN. This option is not available with HYPERMAX OS 5977 or higher.

Unless an *LdevName* is specified, the `symqos` 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 stops.

Note

The `pace` and `priority` parameters are interchangeable, and can be used to identify and set the same parameter.

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 set QoS on `ProdA` to priority 16 to slow the copy of the 1000 volumes.

For SRDF operations, in device group `ProdB`, 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 `ProdB`, 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 `ProdB`, enter:

```
symqos set MIR pace 1 -g ProdDB DEV021
```

Device group QoS query

Examples
To display QoS for a device group, enter:

```
symqos -g dg_test1 query
```

Sample output

```
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.
```

Copy pace settings output display limitations

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.

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 Symmetrix Priority Controls (SPC) **`symqos`** command enables and prioritizes 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 are not allowed a priority setting:

- DATA
- DLDEV
- DRV
- SAVE
- SFS
- Thin
- VAULT
- VDEV

Enable and disable priority of service

Syntax

To enable or disable the priority of service feature for an array, use the following syntax:

```
symqos -pst -sid SymmID enable | disable
```

Examples

To enable host I/O device priority feature for array `1234`, enter:

```
symqos -pst -sid 1234 enable
```

Set host I/O priority

Syntax

When priority of service is enabled, all eligible devices in the array are assigned priority 1 by default.

```
symqos -pst -sid SymmID
[-devs <
SymDevStart:
SymDevEnd|
SymDevName[,<
SymDevStart:
SymDevEnd |
SymDevName>...]>] |
-g DgName] [-std] [-bcv]
set hostio priority Value
```

Options

-pst

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).

Examples

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.

List priority of service

Syntax

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 syntax:

```
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>]
```

Examples

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 symqos command syntax, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

Set bandwidth for Open Replicator (Enginuity 5876 or lower)

Description

The symqos command set and lists port throughput. (These operations were previously performed with the symrcopy command.)

Syntax

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 syntax:

```
symqos -rcopy -sid SymmID
   -dir <# | ALL> [-p <# | ALL>]
   set ceiling <CeilingPercent | DISABLE | NONE>
   list ceiling
```

Options

CeilingPercent

Acceptable values are 0-100, DISABLE or NONE. If the value DISABLE is used, this blocks Open Replicator I/O on the director/port specified. If the value NONE is used, this turns off the ceiling function on this director/port, effectively enabling Open Replicator pace for sessions using this director/port.

Set and reset LRU parameters

Description

LRU ring partitions manage the contention for cache access. The symqos command sets parameters for assigning device groups to LRU rings and manages cache I/O operations.

Syntax

```
set LRU <#,#,#,| ALL> [LdevName]
set LRUname LRUname [LdevName]
reset LRU [LdevName
```

To reset the LRU assignment to the default LRU, use the following syntax:

```
symqos reset LRU LdevName
```

Examples

To assign all devices of device group `ProdDB` to LRU group `2`, enter:

```
symqos -g ProdDB set LRU 2
```

To assign all devices of device group `ProdDB` to an LRU group named `GROUP_2`, enter:

```
symqos -g ProdDB set LRUname GROUP_2
```

Note

Use the `-lru` option of `symcfg list` to list the cache-slot allocation and allocation percentage of a specified LRU cache management group number. To list all the LRUs, specify `ALL`.

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. If devices swap RDF personality, then configure the source and the target the same way.

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 is 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 is 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 takes effect immediately for that director. Any change made to the system default settings takes 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.
- SYMAPI_ACCESS_TYPE_CFGSYM access control privilege is required to the array involved in the action.
- The array must have an SRDF license.

Enable and disable the RDF director workload distribution

Syntax

To enable or disable this feature, use the following syntax:

```
symqos -RA -sid SymmID   enable -io
   disable -io
```

Examples

To enable the workload percentage settings for Synchronous, Asynchronous, and Copy I/Os on array 1234, enter:

```
symqos -RA -sid 1234 enable -io
```

Set the RDF director workload distribution

Syntax

To set the I/O defaults on the SRDF director, use the following syntax:

```
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.

Examples

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
```

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
```

Reset the RDF director workload distribution

Syntax

To rest the workload percentages to the array default values, use the following syntax:

```
reset IO -dir <# | ALL>
```

Examples

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 redistribution 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.

Display RDF workload distribution

RA QoS values

Examples

To display the RDF director workload distribution, enter:

```
symqos -RA list -io
```

View the audit log

Examples

To display the generated logs for the `symqos` commands, enter:

```
symqos list -text
```

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 creation of 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. Unused cache can be donated to other partitions after a specified donation time. Enginuity 5876supports 16 cache partitions.

Create 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.
- Create a cache partition before adding devices to it.

Syntax

To create a cache partition, use the following syntax:

```
symqos -cp -name PartitionName -sid SymmID create -target
TargetPercent -min MinimumPercent -max MaximumPercent -wp
WritePendingLimit -time DonationTime
```

Options

-cp
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.

Examples

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
```

Enable and disable cache partitions

Description

Cache partitioning is enabled and disabled on the array.

Syntax

To enable or disable cache partitions, use the following syntax:

```
symqos -cp -sid SymmID enable | disable
```

Examples

To enable cache partition settings on array 301, enter:

```
symqos -cp -sid 301 enable
```

Device addition to a cache partition

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, 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.

Add devices to a cache partition

Syntax

To add devices to a partition, use the following syntax:

```
symqos -cp -name PartitionName -sid SymmID   add dev SymDevNamesymqos
-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]
```

Options

addall

Use with device groups, device ranges, SRDF groups, and pools. Also used with standard devices (`–std`), BCV devices (`–bcv`), or virtual devices (`–vdev)`

ld

Use with logical devices.

Examples

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:

Note

If the device type is not specified, the default is to add only STD devices.

```
-symqos -cp -name TestP -sid 310 -g Finance addall -bcv
```

To add a thin pool named `Finance` to the cache partition `TestA`, enter:

Note

With Enginuity 5876 and higher, device pools can be added to and removed from cache partitions.

```
symqos -cp TestA -sid 310 addall -pool Finance -thin
```

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 Enginuity versions lower than 5876, DATA devices are blocked.

Modify a cache partition

Syntax

To change the values of a cache partition, use the following syntax:

```
symqos -cp -name PartitionName -ran-sid SymmIDmodify [-target
TargetPercent]
       [-min MinimumPercent
       [-max MaximumPercent]
       [-wp WritePendingLimit]
       [-time DonationTime]
```

The parameter definitions are listed earlier in Create cache partitions on page 461.

Rename a cache partition

Syntax

To rename the cache partition, use the following syntax:

```
symqos -cp -name PartitionName -sid SymmID   rename -new_name
NewPartitionName
```

Examples

To rename cache partition **TestP** on array **310** to **TestA**, enter:

```
symqos -cp TestP -sid 310 rename -new_name TestA
```

Delete a cache partition

Syntax

To delete a cache partition, use the following syntax:

```
symqos delete -cp -name PartitionName
```

When deleting a cache partition, all devices, groups, and pools currently assigned to that partition move back to the default partition.

Remove devices from a cache partition

Description

The `remove dev`, `rmall`, and `remove ld` commands remove device types, a range of devices, a device group, SRDF group, or pool, from a cache partition and returns them to the default partition.

Examples

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
```

Analyze mode for cache partitions

Description

The `symqos analyze` mode determines 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.

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.

Syntax

To set analyze mode for cache partition, use the following syntax:

```
symqos -cp -sid SymmID analyze
```

Options

-usage

Use this option after running in `analyze` mode, as it displays the cache partition settings. and more accurate cache partition target values.

Examples

To set the `–usage` option for cache partition, enter:

```
symqos -cp list -sid 237 -settings -usage Sample output
```

Sample output

```
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
```

Set and display SRDF/A output based on cache partition values

Description

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`.

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.

Sample output

When running the `symqos` command in `analyze` mode, the SRDF/A information shows 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
. . .
```

Preserve empty cache partitions

Description

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).

Syntax

To change this feature and preserve empty cache partitions, use the following syntax:

```
symqos -cp -sid SymmID set empty preserve
```

List cache partitions

Description

Use the `symqos list` command for listing cache partition information for a range of devices, a device group, an SRDF group, an array, or a cache partition name.

Syntax

To list cache partition information using the following syntax:

```
symqos -cp [-name PartitionName]
     list [-devs SymDevStart:SymDevEnd [-all]| -rdfg GrpNum]
     list -g DgName [-std] [-bcv] [-vdev]
     list [-sid SymmID] -settings [-usage] [-v]
```

For complete syntax of the `symqos` command, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

Options

-settings
Lists the general cache partition configuration for the array.

-usage
Includes cache usage information.

Examples

```
symqos list -pst -sid 237 -settings -v
```

Sample output

```
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
```

CHAPTER 14

Virtual LUN Technology (Enginuity 5874 through 5876)

This chapter explains how to change device protection or improve array performance using the `symmigrate` CLI command.

- Virtual LUN Technology overview 470
- Migration requirements by feature 470
- Create virtual LUN migration sessions 472
- Migration to configured space 475
- Migration to unconfigured space 477
- Migrate to a thin pool 478
- Virtual LUN migration management 480

Virtual LUN Technology overview

NOTICE

For VMAX3 arrays running HYPERMAX OS 5977, 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.

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. A virtual LUN migration can be performed online or offline. Use the SYMCLI `symmigrate` command to perform virtual LUN migrations.

A virtual LUN migration may be initiated in the following circumstances:

- Traditional data management: This 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: These involve moving data continuously during its life cycle, either to optimize price/performance or respond to cyclical business changes.

Migration requirements by feature

Table 50 on page 470 lists the hardware, software, disk, and device requirements for migrating devices with virtual LUN technology.

Table 50 Virtual LUN requirements by feature

Enhanced Virtual LUN Technology	Virtual LUN thin-to-thin mobility	Pool-to-pool migrations	VP (Virtual Pool) compression
VMAX arrays running Enginuity 5874, 5875, and 5876	VMAX arrays running Enginuity 5875 and 5876	VMAX arrays running Enginuity 5876	VMAX 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
Supported device types: • Standard VMAX devices, unprotected VMAX devices • Metadevices	Supported device types: • Standard VMAX devices, unprotected VMAX devices • Thin devices	Supported device types: • Thin devices	Supported device types: • Thin devices

Table 50 Virtual LUN requirements by feature (continued)

Enhanced Virtual LUN Technology	Virtual LUN thin-to-thin mobility	Pool-to-pool migrations	VP (Virtual Pool) compression
• FBA, CKD, IBM i 512-byte D910 devices	• Metadevices • FBA, CKD, IBM i 512-byte D910 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 and 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 IBMi thin devices migrate to the target pool.
  - If the target pool contains CKD devices, only the CKD thin devices migrate to the target pool.
- Protected devices cannot be migrated to an unprotected protection type. Migrations of unprotected source devices to unprotected target are supported.
- Migrations are not allowed 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 data may currently be replicated, for example RDF, clone, or snap devices. These devices are blocked from migration. This restriction is for disk group provisioned devices.
- Migrations are not allowed from a thin device to a standard device, or from 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 terminates.

Virtual LUN VP mobility

Virtual LUN VP mobility allows thin devices to be moved between pools. For a given list of devices, data tracks can 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; the rebind option must be done manually if 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. Depending on the VP compression status of the pools, the migration behavior between source and target thin pools behaves as follows :

- 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 for VLUN migration

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 are fully supported.

Create virtual LUN migration sessions

Description

The `symmigrate` CLI provides the options to create and manage migration sessions.

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.

Options

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.

verify
Verifies that a session is in a specified state.

Specify source devices for VLUN migration

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 Device file for source and target pairing for VLUN migration on page 473 and Source file for VLUN migration on page 473.

Device file for source and target pairing for VLUN migration

Description
A device file (*DevFile*) is a text file that contains pairs of source and target devices and one pair per line in the file. Make sure all the devices in the device file match the type of migration specified.

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.

Examples

```
    10      20
    12      22
    14      27
```

For a device to a metadevice, enter:

```
15          28, 29, 30, 31, 32, 33, 34, 39
```

The line continuation character of '\' can be used to wrap lines.

Source file for VLUN migration

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
```

Specify target criteria

Description

Target criteria can be specified as configured or unconfigured space. Specify a target disk group and one of the following target protection types (unless using pairs in a file):

- Unprotected
- RAID1
- RAID 5 (3+1 or 7+1)
- RAID 6 (6+2 or14+2)

The target protection type and disk group is assumed when using a device file.

Options

The `symmigrate` command supports the following options:

-tgt_pool
Specifies this target type as the destination for a thin migration.

-pool *PoolName*
Specifies the destination pool.

Establish a migration session

Use the `symmigrate establish` command to create the migration session and begin migrating source devices to the new target location. Provide the target protection type and disk group number, unless using a device pair file or thin devices. 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. 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 support up to 20,000 devices per migration.

The migration control operation operates on the metahead only; 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 as long as the devices are matched in size and quantity to the meta members. 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.

Validate a migration session

The `symmigrate validate` command is an optional command that can be used in two ways:

- As a simple positive/negative test to determine if a migration will succeed. This is denoted by the return value of the command.
- 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 and to modify the resulting file if necessary.

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.

Migration 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.

Migrate to configured space using a device file

Syntax

Use the following syntax to create a migration session using a device file:

```
symmigrate [-v][-i Interval] [-c Count] [-noprompt]
-name SessionName -file DevFile -sid SymmID validate
establish
```

Examples

```
symmigrate validate -sid 123 -f input.txt -name DevMigration
symmigrate establish -sid 123 -f input.txt -name DevMigration
```

Migrate to configured space specifying target criteria

Syntax

Use the following syntax to perform a migration to configured space with target criteria:

```
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
```

Options

-name *SessionName*
Indicates a session name when validating or establishing a session.

-file *SrcFile*
Specifies a file as the source device.

-sid *SymmID*
Specifies the array ID.

-sg *SgName*
Specifies a storage group as the source device.

-g *DgName* [-bcv | -tgt]
Specifies a device group as the source device, with an optional pointer to the BCV or TGT list.

-tgt_config
Tells the array to look for configured space.

-tgt_dsk_grp *DskGrp* | *name:DskGrpName*
Identifies the disk group number or name for the migration. The source devices will be in that disk group when the migration is complete.

-tgt_unprotected
Specifies the devices will be unprotected when the migration completes. This option is only valid if the source devices are unprotected.

-tgt_raid1
Specifies RAID 1 protection.

-tgt_raid5
Specifies RAID 5 protection with either `-tgt_prot 3+1` or `7+1`.

-tgt_raid6
Specifies RAID 6 protection, with either `-tgt_prot 6+2` or `14+2`.

Error reporting for VLUN migration (configured space)

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:

- An invalid configuration
- A local or remote replication or migration session
- Host visible
- A capacity, emulation, protection type, or block size mismatch
- Reserved

Use the `symmigrate validate` command to produce a file that indicates which devices the system identifies as the target at the time the `symmigrate validate` command is run.

Example of a failed migration session
For example, a device input file named `test1` contains:

```
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. Note that the -v (verbose) option triggers 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
```

Migration to unconfigured space

Syntax

Use the following syntax to migrate 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
```

Options

-name *SessionName*
: Indicates a session name when validating or establishing a session.

-file *SrcFile*
: Specifies a file as the source device.

-sid *SymmID*
: Specifies the array ID.

-sg *SgName*
: Specifies a storage group as the source device.

-g *DgName* [-bcv | -tgt]
: Specifies a device group as the source device, with an optional pointer to the BCV or TGT list.

-tgt_unconfig
: Tells the array to look for unconfigured space.

-tgt_dsk_grp *DskGrp* | *name:DskGrpName*
: Identifies the disk group number or name for the migration. The source devices will be in that disk group when the migration is complete.

-tgt_unprotected

Specifies the devices will be unprotected when the migration completes. This option is only valid if the source devices are unprotected.

-tgt_raid1
Specifies RAID 1 protection.

-tgt_raid5
Specifies RAID 5 protection with either `-tgt_prot 3+1` or `7+1`.

-tgt_raid6
Specifies RAID 6 protection, with either`-tgt_prot 6+2` or `14+2`.

Example
To migrate devices in a device group `-tgt` list to unconfigured space, with RAID 5 3+1 protection, enter:

```
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 for VLUN migration (unconfigured space)

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:

- An invalid configuration
- A local or remote replication or migration session
- Host visible
- a capacity, emulation, protection type, or block size mismatch
- Reserved

Example of a successful migration
The device input file contained:

```
0A33
```

The following output shows the message from a successful **`symmigrate establish`** for the unconfigured devices. Note that the `-v` (verbose) option 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.
```

Migrate to a thin pool

Description
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`.

Note

Remove the source pool from FAST control before moving allocations to the target pool.

Syntax

Use the following syntax to migrate 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
```

Options

-name *SessionName*
: Indicates a session name when validating or establishing a session.

-file *SrcFile*
: Specifies a file as the source device.

-sid *SymmID*
: Specifies the array ID.

-sg *SgName*
: Specifies a storage group as the source device.

-g *DgName* [-bcv | -tgt]
: Specifies a device group as the source device, with an optional pointer to the BCV or TGT list.

-src_pool *PoolName*
: 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 *PoolName*
: Specifies the target pool for a thin migration.

View a thin migration

Options

-detail
: Displays migrations sessions with both flag and target descriptions.

Examples

To display the migration sessions currently on array 266, enter:

```
symmigrate query -name sample_test -sid 266
```

Sample output

```
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.
```

Ouput using the `-detail` option.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.

```
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.
```

Virtual LUN migration management

This section describes how to use the `symmigrate list`, `verify`, `query`, and `terminate` options.

Note

If a session exists for two weeks without being listed or queried, it is automatically removed and will not be returned by a `symmigrate query` or `list` command.

The array device lock is held on the source device only when the `establish` and `terminate` operations are in progress. The lock is released when the migration enters the Sync In Progress state and remains released until the session ends. When the session is terminated, the lock will be taken again.

For devices in disk group migrations, up to 16 concurrent migration and configuration change sessions can be run. For thin devices in thin pool migrations, the limit is 128.

List VLUN migration sessions

Syntax

```
symmigrate [-v] [-i Interval] [-c Count]
list [-sid SymmID] [-names] [-detail]
```

Examples

To display the migration sessions currently on array 266, enter:

```
symmigrate -sid 266 list
```

To display the migration sessions currently on array 266 along with source and target information for each session, enter:

```
symmigrate -sid 266 list -detail
```

Sample output

```
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.
```

Output with source and target information for each session:

```
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.
```

Query VLUN migration sessions

Description

Use the `symmigrate query` command to display the status of a migration session.

Syntax

```
symmigrate [-v] [-i Interval] [-c Count]
            -name SessionName-sid SymmID
   query [-detail][-summary]
```

Options

-detail
Provides tables listing all sessions and their current session states.

-summary
Includes a summary listing of session states and the number of sessions in each state to assist with interpreting the command output. This option provides 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. Note that the `Total` field that shows total invalid tracks is formatted differently when `-summary` is specified.

-i *Interval*
Specifies the migration interval.

-c *Count*
Specifies the migration count.

Example

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
```

Note

`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.

Verify VLUN migration sessions

Description

Use the `symmigrate verify` command to determine if a migration session is in a specific state. A virtual LUN migration session is complete when it is in the migrated state.

Syntax

```
symmigrate -sid SymmID -name SessionName[-i Interval] [-c Count]
   verify [-createinprog | -syncinprog | -synchronized
           -migrateinprog | -migrated | -failed | -invalid]
          [-summary]
```

Options

-createinprog
Verifies that migration device pairs are in the CreateInProg state.

-syncinprog
Verifies that migration device pairs are in the SyncInProg state.

-synchronized
Verifies that migration device pairs are in a synchronized state.

-migrateinprog
Verifies that migration device pairs are in the MigrateInProg state.

-migrated
Verifies that migration device pairs are in a migrated state.

-failed
Verifies that migration device pairs are in a failed state.

-invalid
Verifies that migration device pairs are in an invalid state.

-summary
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`

Example

To verify whether a migration session named `migrate1` is in the migrated state, enter:

```
symmigrate -sid 258 -name migrate1 verify -migrated -i 30

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.
```

Terminate VLUN migration sessions

Description

Use `symmigrate -terminate` to terminate a migration session. 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.

Syntax

To terminate a session, use the following syntax:

```
symmigrate -name SessionName-sid SymmID>
   terminate [-noprompt]
```

View the audit log

Use the `symaudit` command to view the audit log for the migration session. For information about `symaudit`, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

For any control command, the API logs the start and end of the operation and any errors that may occur. Table 51 on page 484 lists the information that is saved in the audit log based on the target selection type.

Table 51 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

PART 2

Querying and Reporting

This section describes the SYMCLI query commands and various tools for monitoring and managing storage array performance.

Chapters include:

Chapter 15, "Configuration Query Operations"

Chapter 16, "Statistics"

Chapter 17, "Change Tracker (Enginuity 5876 or lower)"

Chapter 18, "Optimizing Array Performance (Enginuity 5876 or lower)"

Chapter 19, "Managing Time Windows (Enginuity 5876 or lower)"

Chapter 20, "Managing Network IPsec (Enginuity 5876 or lower)"

Chapter 21, "Events and Logs"

Chapter 22, "XML Structured Output"

CHAPTER 15

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 overview 488

Configuration data overview

The Solutions Enabler SYMCLI provides various commands that are used to query different levels of the storage environment. These levels include:

- SCSI-level — Returns data 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 —Returns 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. Other details include vendor specific configuration information, ports, flags, adapters, Data at Rest Encryption data, pool information, and environment data.
- Device-level — Returns device level data on locally or remotely attached arrays. 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 such as 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. This command returns SCSI-level data for EMC arrays, StorageWorks, or HDS devices.

The `syminq` command can return a list all of the HBAs on the local host. Using the options available, the scope of this request can be limited to 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.

Returned results can be limited to devices with target mapping information (only devices mapped through fibre HBAs using the `-mapinfo` option), or to 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`.

The *EMC Solutions Enabler SYMCLI Command Reference Guide* describes the various options available with the command.

Note

Data returned from issuing any `syminq` command is not stored in the configuration database.

SCSI device list

Examples

To list all devices listed by physical device name, enter:

```
syminq
```

Sample output

```
      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

Examples

To list devices by array ID, enter:

```
syminq -symmids
```

Sample output

```
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
. . .
```

List of devices without capacity

Examples

To list devices without issuing a SCSI READ CAPACITY, enter:

```
syminq -symmids -nocap
```

Sample output

```
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
```

List of devices with WWN

Options

-colon

Use this option to return a list of devices with WWN for each device using colons as spacers.

Examples

To list the WWN for each device, enter:

```
syminq -wwn
```

Sample output

```
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
```

Output with `-colon` option:

```
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
```

Device list in pdevfile format

Examples

To list device names in a format suitable for use as `pdevfile`:

```
syminq -pdevfile
```

Sample output

```
# 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
```

List HBA information

Options

-fibre
Lists only the Fibre HBAs.

-scsi
Lists only the SCSI HBAs.

-iscsi
Lists only the iSCSI HBAs.

-snia
Returns 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 is obtained.

Examples

To obtain a list of the local host's HBAs, enter:

```
syminq hba
```

Note

HBA returned data will vary slightly when the `-scsi` or `-iscsi` option is specified.

Sample output

```
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 : 00000100000000001000000000000000000000000000000000000000000000000000
      Active FC4 types    : 00000100000000001000000000000000000000000000000000000000000000000000
      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 : 00000100000000001000000000000000000000000000000000000000000000000000
      Active FC4 types    : 00000100000000001000000000000000000000000000000000000000000000000000
      Max frame size      : 2048
```

List device mapping information

Syntax

To list mapping information, use the following syntax:

```
syminq -mapinfo[PdevName]
         [-sym[-powerpath]|-hds|-storworks]
         [-cache | -nocache][-colons][-winvol]
```

Examples

```
syminq -mapinfo
```

Sample output

```
               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
```

List device identifiers

Description

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`.

Examples

To list the VMS device identifiers, enter:

```
syminq -identifier vms_id
```

Sample output

```
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

The `symcfg` command provides arguments and options to report specific array-level components. 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 specified 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 — A list of 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.

List arrays

Description

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. The `symcfg` command lists of all the accessible arrays by SID including the model number and the number of accessible devices. Use the SID with other command options to obtain configuration information for the array directors and devices.

Examples

To list all array IDs connected to a host, enter:

```
symcfg list
```

To obtain a detailed list for a specific array, using the verbose option (`-v`), enter:

```
symcfg list -v -sid 2300
```

Sample output

Output listing all arrays connected to a host.

```
                          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       VMAX40K     5876       98304        20     32088
000000006206 Remote      VMAX40K     5876       32768         0      4139
000187900035 Remote      VMAX40K     5876        8192         0      1314
000187900041 Remote      VMAX40K     5876        8192         0       960
000190300174 Remote      VMAX40K     5876       16384         0       335
000190300175 Remote      VMAX40K     5876       16384         0       914
000190300184 Remote      VMAX40K     5876       16384         0       417
```

Output listing details (`-v`) for array `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. Refer to Setting compatibility on page 33 for information on setting output reporting formats for older SYMCLI versions.

List array bay locations (Enginuity 5875 or 5876)

Example

To list the bay names and location information of array `166`, enter:

```
symcfg list -bay_location -sid 166
```

Sample output

Note

This information is only displayed if an EMC Customer Engineer created it, and if the array is running Enginuity 5875 or 5876.

```
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
```

Set the VMAX 10K and 40K LED door feature (Enginuity 5876 only)

Description

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. This feature is not supported on VMAX 20K arrays.

Syntax

To set the LED by array ID or bay name, use the following syntax:

```
symcfg
         set -led [blue_flash | on] [-sid SymmID]
         set -led [slow_blink | on] [-bay_name Bay Name
```

Options

-led

Indicates that the system or drive bay LED will be controlled.

blue_flash

Sets the LED to blue flash mode. 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 mode and are not be set to this mode when the `blue_flash` operand is specified. 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.

slow_blink
Sets the LED to slow blink mode. 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.

on
Turns LED on. 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.

-bay_name
Specifies the system bay or drive bay to control the LED. Where *Bay Name*must be specified as:

- *SB-x* where *x* is a system bay drive.
- *DB-xy* where *x* is a drive bay index and *y* is the bay side (A, B, C, or D).

Examples
To set the LED mode to `blue_flash` for array `188`, enter:

```
symcfg set -led blue_flash -sid 188

LED state successfully changed.
```

To set the LED mode to `slow_blink` for system bay `SB-2` on array `188`:

```
symcfg set -led slow_blink -bay_name SB-2 -sid 188

Bay SB-2 LED successfully set to slow blink.
```

If actions fail, these command displays a descriptive error message.

List array bay LED states

Enginuity 5876 or lower
Options

-bay_info
Lists bay information. Equivalent to the `-bay_location` option, but more description of the bay functionality.

Examples

```
symcfg list -bay_info
```

Sample output

```
    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      :
```

Enginuity 5875 or 5876

Options

-v

Displays the LED state as a composite of all bays.

Examples

```
symcfg list -sid 341 -v
```

Sample output

```
. . .
Maximum number of hypers per disk    :   512
Composite door LED status            : On
Number of Powerpath Devices          :   0
. . .
```

Output values for LED status are:

- BLUE_FLASH
- FAST_BLINK
- SLOW_BLINK
- ON
- OFF
- N/A

Enginuity 5874 or lower

Examples

```
symcfg list -sid 237 -v
```

Sample output

Note

The `Composite door LED status` displays as `N/A`. The LED status display requires Enginuity 5875 or higher. All other LED state display functionality requires Enginuity 5876 or higher.

```
. . .
Maximum number of hypers per disk    :   512
Composite door LED status            : N/A
Number of Powerpath Devices          :   0
. . .
```

Data at Rest Encryption (Enginuity 5875 or higher, and HYPERMAX OS 5977)

For arrays running Enginuity 5875, 5876, and HYPERMAX OS 5977, Data at Rest Encryption (D@RE) provides hardware-based, on-array, back-end encryption for VMAX family arrays. Back-end encryption protects 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 VMAX Family (10K, 20K, 40K) Product Guide* and the *EMC VMAX3 Family 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 provides 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.

List data encryption status

Options

-v

Lists array details including data encryption status.

Examples

To list data encryption status (using `-v` option) for array 343, enter:

```
symcfg list -sid 343 -v
```

Sample output

If encryption has not been turned on at install time, the `symcfg -v` command displays `Symmetrix Data Encryption` as `Disabled`.

If encryption is not supported on the array, the `symcfg -v` command displays `Symmetrix Data Encryption` as `N/A`.

```
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
. . .
```

List Data encryption key (DEK) audit log

Description

Every key management event generates an entry in the array audit log, including initial DEK configuration information.

Example

```
symaudit list -text -sid 012 -function_class Security
```

Sample output

```
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:
0FC78B4ECA29
```

Post-drive replacement DEK audit log

Description

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.

Example

The following output lists this information for array `012`:

```
symaudit list -text -function_class Security -sid 012
```

Sample output

```
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
```

View application registrations with array access

Examples

For example, to list all the applications that have host to array connection for array `282`, enter:

Note

Omit the `-sid` option list connections for all arrays.

```
symcfg list -applications -sid 282
```

Sample output

Note

For HYPERMAX OS 5977, if Embedded Management is configured on the VMAX array, the node name and IP address listed in the output is the nodename and IP address identified by the NAT gateway, and not the internal identity of the eManagement Guest. For more information on eManagement refer to the *EMC VMAX3 Family Product Guide*.

```
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).
```

List host connections to arrays

Options

-connections

Returns the host connections to the array. Only hosts that have at least one registered application are listed.

Examples

To list the host connections to array `097` enter:

Note

To view the host connections for all arrays, omit the **-sid** option.

```
symcfg list -connections -capacity -sid 097
```

Sample output

Note

When displaying all arrays, the output is sorted according to each array

```
  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
```

List host connections sorted by host names

Options

-ipv6
Displays a layout that does not truncate node names or addresses.

Examples

To list all of the host connections sorted by host names for array `097`, enter:

```
symcfg list -connections -sorthost -sid 097
```

Sample output

```
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
```

Supported director configuration types

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

Note

Director types IM and EDS are only supported by HYPERMAX OS 5977 or higher.

List director configuration data

Examples

To list configuration and status information about all directors on array 00019490004, enter:

```
symcfg list -dir ALL -sid 064
```

Note

Director types IM and EDS are only supported by HYPERMAX OS 5977 or higher.

Sample output

```
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
```

List port flag information

Syntax

To list port flag information, use the following syntax

```
symcfg [-sid <SymmID>][-offline]
    list -port -free
    <[-slot <#>] [-dx | -fa | -fcoe | -re | -rf | -se]
     [-speed <#>] |
    -dir <#>>
```

Options

-free
Lists the ports that are not associated with a director emulation.

-dx
Lists only DX (external) directors.

-fa
Lists only FA (Fibre) directors.

-fcoe
Lists only FCOE (Fibre Channel Over Ethernet) directors.

-re
Lists only RE (RDF Gig-E) directors.

-rf
Lists only RF (RDF Fibre) directors.

-se
Lists only SE (Gig-E) directors.

Support for multiple cores and ports

HYPERMAX OS 5977

For arrays running HYPERMAX OS 5977, all director types can of support multiple cores. The actual number of cores assigned to a director are statically configured. Use the `symcfg list - dir` command to view the number of cores assigned to a given director. Also with HYPERMAX OS 5977, all director types are capable of supporting a variable number of ports. Solutions Enabler supports 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.

Enginuity 5876 and lower

For arrays running Enginuity 5876 or lower, 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 more IOPS or front-end connectivity is needed, add another emulation instance and its one or two additional ports.

List director information by type

Examples

To list all front-end directors (`-FA`) for array `005`, enter:

```
symcfg list -FA ALL -sid 005
```

Note

Director types, `-CA` and `-SA`, are no longer supported. Special format displays such as those previously produced by the specification of `-FA` or `-SE` options are no longer supported. Two new director types are supported for Infrastructure Manager (IM) and Enginuity Data Services (EDS).

Sample output

```
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
```

List director details by name and type

Options

-v (verbose)

Lists details about a specific director and type.

Examples

To display information about the Gig-E director `5G` on array `064`, enter:

```
symcfg list -SE 5G -sid 064 -v
```

Sample output

```
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
```

List director port data

Options

-port

Lists ports that are online or offline on SA, FA, SE, or RA directors.

-p ‹#›

Restricts output to a specific port number.

Examples

To list port status for all `SE` directors on array `230`, enter:

```
symcfg list -port -SE ALL -sid 230 -detail
```

Sample output

```
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
```

List addresses of devices mapped to directors

Options

-address

Identifies the address information for devices accessible through specific directors from a host-based view of the storage environment.

Examples

To list the address information for all director types on array `064`, enter:

```
symcfg list -dir ALL -sid 064 -address
```

To list address information for port `10` on director FA `2E`, enter:

```
symcfg list -sid 075 -FA 2E -address -p 10
```

Sample output

For all director types:

```
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 port `10` on director type `FA`:

```
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)
```

Note

For arrays running HYPERMAX OS 5977, when LUN information is not available, VBUS TID, and LUN address are reported as N/A.

List the next available device address

Options

-available

Returns the next available address that can be used for a device.

Examples

To list next available address on array `097`, enter:

```
symcfg list -dir all -address -available -sid 097
```

Sample output

Note

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.

```
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 *
```

Take RA directors offline

Examples

To take `RA-12` for array `097` offline, enter:

```
symcfg offline -RA 12 -sid 097
```

Bring RA directors online

Examples

To bring RA director `12` online for array `097`, enter:

```
symcfg online -RA 12 -sid 097
```

Take front-end director ports offline

Examples

To take port `4` of `SA-12` in array `097` offline, enter:

```
symcfg offline -SA 12 -P 4 -sid 097
```

NOTICE

Do not turn off the only connection from the host to the array, otherwise another host that has connection to the array must be used to bring a director port back online.

Bring front-end director ports online

Examples

To bring port `4` of `SA-12` in array `097` back online, enter:

```
symcfg online -SA 12 -P 4 -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.) Use the `symcfg list -lockn` command, to list all locks on one or all arrays or just the locks targeted to specific operations.

List all array locks

Examples

To return a list of all host-visible arrays (local and remote), along with details about all array exclusive locks, enter:

```
symcfg list -lockn all
```

Sample Output

The returned list contains three local arrays that have no known locks, as specified by the `N/A` values. Remote array `000187900039` has an exclusive lock number `15` for a configuration change activity (`ConfigChg`).

```
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,

List lock number details

Examples

To return a list of all host-visible arrays (local and remote), and details about lock `0`, enter:

```
symcfg list -lockn 0
```

Sample output

Note

If an array is holding a lock other than 0, the output still returns a lock number of `N/A`.

```
                       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
```

List lock details

Options

-v

Lists extended lock information about the application and host that owns the lock

all
Lists locks for all arrays.
#
Lists specific lock number.
RDF
Lists SRDF locks only.
RDFA
Lists SRDF/A locks only.
SRDF_MSCS
Lists SRDF MSCS locks only.
GNS
Lists GNS locks only.

Examples
To list information about lock `23`, enter:

```
symcfg list -lockn 23  -sid 190
```

To return verbose information about lock `15`, enter:

```
symcfg list -lockn 15 -sid 207 -v
```

Sample output
Output for information on lock `23`.

```
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
```

Output for verbose information on lock `15`.

```
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
```

Release an array lock

Description
Releasing a lock on an array is available but not recommended unless it is confirmed that the lock is stranded.

Examples
To release an external lock `0`, which has confirmed to be stranded on array `097`, enter:

```
symcfg -sid 097 -lockn 0 release
```

Enginuity 5874 or higher

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 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. Using the `symconfigure` or the `symmigrate` command, abort the correct session that owns the lock.

For information about device external locks that only target specific devices, refer to Device external locks on page 544.

List all LRU cache management groups

Examples

To view a list of all LRUs for array `6196`, enter:

```
symcfg list -lru all -sid 6196
```

Sample output

```
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
```

Optimized Read Miss (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 higher versions 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.

Set and show Optimized Read Miss Mode

Syntax

To set the Optimized Read Miss mode for a single device or a list of devices, use the following syntax:

```
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]
```

Examples

To show Optimized Read Miss status for device `01F2`, enter:

```
symdev show 01F2 -sid 46
```

Sample Output

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 the current Enginuity version, the Optimized Read Miss mode displays as N/A.

```
. . .
    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 using storage groups (`symsg`), device groups (`symdg`), or composite groups (`symcg`).

Network configuration file

The Solutions Enabler configuration file `netcnfg` is used by the SYMCLI client library to specify attributes of a remote server where array management operations should 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.

List SYMAPI services

Description

The `symcfg -services list` command validates the syntax of the `netcnfg` file entries and displays the configured network services available for use by the SYMAPI client connection.

Examples

```
symcfg list -services
```

Sample output

```
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    vmax1.mmcs1.xyz.com               2707 SECURE
    ord_guest    Ordered  TCPIP    vmax1.mmcs1.com                   2707 SECURE
    ord_guest    Ordered  TCPIP    vmax1.mmcs2.com                   2707 SECURE
    bal_guest    Balanced TCPIP    vmax1.mmcs1.xyz.com               2707 SECURE
    bal_guest    Balanced TCPIP    vmax1.mmcs2.com                   2707 SECURE
```

Pairing Method descriptions:

- Single — Single entry service name.
- Ordered — The SYMAPI client library first attempts a client/server session with the server named as the first of the two entries in the `netcnfg` file. If that attempt fails, the library tries the second entry.
- Balanced — The SYMAPI client library applies a random number to select the first entry to attempt a client/server session. So it may select the first or the second entry in the `netcnfg`file .

List memory board information

Description

Memory board information includes the number of boards, the slot number, and the capacity information in MBs.

Examples

To view the available memory board information for all arrays, enter:

```
symcfg list -memory
```

Mainframe CU image

The `symcfg` command is 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 creates 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.

List CU images

Examples
To list CU images for all arrays, enter:

```
symcfg list -cuimage
```

Show specific CU image details

Examples
To list CU image `0` for subsystem ID `640`, enter:

```
symcfg show -cuimage 0 -ssid_num 640
```

List Enginuity patches

Examples
To list all of the Enginuity patches installed on a array `207`, enter:

```
symcfg list -upatches -sid 207
```

Sample output

```
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

Use the `list -env_data` option to list the status of the major hardware modules including fans, power supplies, drive enclosures, and link control cards. An array ID must be specified when querying for environmental data.

List all environment data on array

Examples
To list an overall status for all environmental components on array `150`, enter:

```
symcfg -sid 150 list -env_data
```

Show environmental data on array component

Examples

To return a detailed status for each environmental component for `SystemBay` on array `150`, enter:

```
symcfg -sid 150 show -env_data SystemBay
```

List environmental data for specific service state

Options

-service_state

Returns data for possible service states: `degraded`, `failed`, or `normal`. To list all service states except one prefix the service state value with `not`, such as –`service_state notfailed`

Examples

To list the environmental data for array `150` with a service state of `failed`, enter:

```
symcfg -sid 150 list -env_data -service_state failed
```

Note

Returned data only contains information about the bay containing the failure, and the components in the failed state.

Environmental data output display (HYPERMAX OS 5977 and higher)

Differences in hardware configurations between arrays running HYPERMAX OS 5977 and arrays running Enginuity 5876 or lower requires changes to the Environmental Data reporting. Changes are made to the output display for `symcfg list -env_data -v` command to support these differences. The output 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.

Listing array environmental data example (HYPERMAX OS 5977 or higher)

Examples

To list environmental data details for array `64`, enter:

```
symcfg list -env_data -v -sid 64
```

Sample output

Output for an array running HYPERMAX OS 5977 containing a system bay and at least one drive bay:

```
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
```

List device pools

Syntax

```
symcfg  [-sid SymmID] [-offline] [-mb | -gb | -tb]
       [-i Interval] [-c Count]
   list [-pool [-snap][-rdfa_dse [-rdfg GrpNum]][-thin]
          [-fba] [-ckd] [-ckd3390] [-ckd3380] [-as400] [-all] [-v]]
```

Options

-i *Interval* -c *Count*
: Checks status of the pool(s) continuously for a certain period of time.

-v
: List details of each pool in the returned data set. It is equivalent to using the *GrpNum* 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 is 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` *GrpNum* 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 thin device pools only.

-all
Includes both enabled and disabled devices in the calculations for `Free Tracks`and `Full %`. Otherwise, only enabled devices are included. When specifying 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.

Examples

To list details about all thin pools in array `087`:

```
symcfg list -pool -sid 087 -thin -all -detail
```

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.

Sample output

```
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
```

Note

The pool `State` indicates whether there are any devices enabled in the pool.

Show thin pool rebalancing

Syntax

To show pool rebalancing parameters, use the following syntax:

```
symcfg show -thin -pool PoolName -detail -all -sid SymmID
```

Note

Pool rebalancing is for thin pools only and is not applicable to Snap or SRDF/A DSE pools.

Examples

To show pool rebalancing parameters for thin pool `Mig_trg2` on array `432`, enter:

```
symcfg show -pool Mig_trg2 -thin -sid 432 -all -detail
```

Sample output

```
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.

Table 52 Pool Rebalancing Parameters

Pool Rebalancing Parameters	Description
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 default is `256`.

List feature registrations and usage data

Description

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.

Options

-v
Displays usage information.

-class
Llimits the feature display to a specified class of features.

Examples

To list feature class registrations for `Local Replication` class on array `341`, enter:

```
symcfg list -feature -sid 341 -class Local Replication
```

Sample output

```
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 is many *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, a list of all available devices can be displayed. The listed device data is used to obtain configuration and status information. This information identifies back-end information for the device's disk directors and corresponding hypervolumes, and their mappings to disk drives.

Device types

SYMCLI defines and configures devices for numerous specialized roles defined as *device types*. Each device type has specialized characteristics that enables a device to participate in various SYMAPI operations.Table 53 on page 520 and Table 54 on page 520 describe the device types of the DMX and VMAX storage environment.

Device types - HYPERMAX OS 5977

Table 53 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

Table 54 Device types - Enginuity 5876 and lower

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.

Table 54 Device types - Enginuity 5876 and lower (continued)

Device type	Description
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.
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

Table 54 Device types - Enginuity 5876 and lower (continued)

Device type	Description
	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.*

List array device information

Description

Use the following 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.

Syntax

To list host-visible devices, use the following syntax:

```
sympd [-offline] [-sid <SymmID>] [-v]
        list [-resv]
        list [-SA <#|ALL>] [-p <#>] [-scsi] [-fibre]
          [-escon] [-ficon] [-gige] [-iscsi_port <#>]
          [-powerpath] [-vcm | -aclx] [-pdevfile] [-cyl]

     sympd [-offline] [-sid <SymmID>] [-v]
        list [-DA <#|ALL>] [-interface <#|ALL>]
          [-disk <#|ALL>] [-hyper <#|ALL>]
          [-spindle]
        list [-DX <#|ALL] [-spindle]
        list [-vm]
```

Options

-sid
Limit the device output to a specific array.

-DA
List host-visible array devices that match a specific DA.

-DX
Lists host-visible array devices that match a specific DX.

-interface
Lists host-visible array devices that match a specific interface.

-disk
Lists host-visible array devices that match a specific disk.

-hyper
Lists host-visible array devices that match a specific hyper-volume.

-spindle
Includes Spindle ID Information.

-vm
Displays valid virtual machine names on VMware ESX server environments.

-SA
List the host-visible array devices that match a specific front-end director number.

-P *#*
List the host-visible array devices that match a specific front-end director port number.

-scsi
List the host-visible array devices that are mapped to SCSI front-end directors.

-fibre
List the host-visible array devices that are mapped to Fibre front-end directors.

-ficon
List the host-visible array devices that are mapped to FICON front-end directors.

-escon
List the host-visible array devices that are mapped to ESCON front-end directors.

-gige
List the host-visible array devices that are mapped toGig-E front-end directors

---

Note

The `sympd` command only provides a limited amount of filter options. Use the `symdev list pd` command for additional filter options on physical devices. Refer to Filter device list device data on page 540 for more information.

Examples

To list host-visible array devices that match a front-end director `07E` for array `064`, enter:

```
sympd list -SA 7E -sid 064
```

Sample output

Array device names are listed in the `Sym` column.

```
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
```

Report spindle ID information

The following commands report spindle ID information:

- sympd show
- sympd list -v
- symdev list
- symdev show

Note

For Enginuity versions 5875 or lower, the specified/returned Spindle ID does not represent an actual Enginuity entity, but is a string representation of a unique hexadecimal number, which may be used as an input parameter for `sympd show` commands.

---

sympd show

- For all but unprotected devices, the spindle ID reports as part of the `RAID Group Information` section.
- For unprotected devices, the spindle ID reports in both the `RAID Group Information` section and the `Back-end Disk Director Information` section.
- For non-RAID devices (VAULT devices), spindle IDs report in the `Back-end Disk Director Information` section.

sympd list -v
Reports spindle ID information as part of both the back-end disk director Information section and the RAID group Information section.

symdev list

- Lists host-visible array devices for the following specified values:
  - DA (`-da`)
  - DX (`-dx`)
  - interface (`-interface`)
  - disk (`-disk`)
  - hyper-volume (`-hyper`)
- When specified along with the `-spindle` option, the reports Spindle ID information.

symdev show

- For all but unprotected devices, the spindle ID reports as part of the `RAID Group Information` section.
- For unprotected devices, the spindle ID reports in both the `RAID Group Information` and the `Back-end Disk Director Information` section.
- For non-RAID devices (VAULT devices), spindle IDs reports in the `Back-end Disk Director Information` section.
- Provides similar output as the `sympd` command, but includes all array devices and lists them by array device names.

External spindle devices

Federated Tiered Storage (FTS) allows for 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

Use the `symdev list` command with the `-external` option to list information about eDisks (external spindles).

Note

FTS is supported only with Enginuity 5876 and HYPERMAX OS 5977.

HYPERMAX OS 5977
FTS is limited to the following:

- 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.
  - A device status of *Not Ready* for a Data Domain device indicates that the device is *Not Ready* because it is specified as the target of an image refresh.
- External provisioning for XtremIO (known as FAST.X)

List external spindles

Syntax
To list external spindles, use 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]
```

Options

-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, and is abbreviated as `-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, and is abbreviated as `-ext`.

-encapsulated
Lists encapsulated devices, and abbreviated as `-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.

Note

The `symdisk list` command also provides options for listing external spindle information. For more information, also refer to External spindle information on page 553.

List RAID group information

Description

The `symdev` command indicates whether a rebuild or copy is in progress for a member of a device's RAID groups. The `RAID Group Information` section includes an additional indicator under the `Status` column, which displays when a specific member is undergoing one of the following:

- Rebuild — (R)
- Member copy — (C)
- Failed state — (F)

Examples

To list if a rebuild or copy is in progress for a member of the RAID groups for device `1234`, enter:

```
symdev show 1234 -sid 123
```

Sample output

```
. . .
  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
                }
```

Change the device state

Syntax

To change the device state, set or reset the hold bit on a device, or relabel a device, use the following 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]
```

Change device state using filename

Syntax

To change the device state of multiple devices listed in a file, use the following syntax:

```
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 arrays running HYPERMAX OS 5977 or higher `-SA | FA ALL` options are supported. The `[-SA | FA <#> -P <#>]` options are not supported.

View device service states

Description

The `symdev list` and `symdev list pd` commands provide options that return data that is limited to a specific service state.

Options

-service_state

Returns list of devices one of the following states:

- Normal
- Failed
- Degraded — Indicates if one or more mirror positions of the protected device are not ready.

To list service states not in a particular state use "`not`" such as, `notfailed`.

Examples

To list all devices on array `097` that are in a `normal` state, enter:

```
symdev list -service_state normal -sid 097
```

to list all devices on array `097` that are `notfailed` service state, enter:

```
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.)

List device emulation types

Examples

To return a list of configured devices by emulation type for array `3139`, enter:

```
symdev list -inventory -sid 3139
```

Sample input

```
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
```

Show device details

Description

Due to local RAID storage management differences between Enginuity and HYPERMAX OS several changes have been made to the `symdev show` command to present device details across all supported microcode levels. These changes present a common format for Mirror and RAID information regardless of RAID type and microcode version.

Examples

To details for device `8423` on array `584`, enter:

```
symdev show 8423 -sid 584
```

Sample output

Note

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.

```
. . .
    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
```

Output values for vault devices (Enginuity 5876 or lower)
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.

Show 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

HYPERMAX OS 5977

HYPERMAX OS 5977 supports 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

Enginuity 5876 supports TimeFinder VP Snap functionality. VP Snap leverages 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.

Note

Refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide* for more information on the VP Snap feature and VSE.

Show disk geometry details

Examples

To show disk geometry for device `0016` on array `516`, enter:

```
symdev show 0516 -geometry -sid 516
```

Sample output

Note

If the array-wide setting is turned on, the geometry at the individual device level can be defined which overrides the array-wide setting. In addition, for non-FBA devices, the output displays `N/A` with the `-v` and the `-geometry` options.

```
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):
```

Output display

The `Effective Device Geometry` field has the following 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.

List DATA and SAVE devices

Syntax

To list DATA and SAVE pool device details, use the following syntax:

```
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...]]
```

Note

When the `symcfg list` command is used to return pool information, 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.

Options

-all

Includes both enabled and disabled devices in the calculations of `Free Tracks` and `Full %`, and the `Full %` is based on the `Usable Tracks`. Otherwise, only enabled devices are included. 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.

-devs

Filters for specific devices or device range.

-nonpooled
Filters for nonpooled devices.

Examples
To return a list of DATA devices for array `237`, enter:

```
symcfg list -sid 237 -datadev
```

List thin device information

Description
In addition to listing thin devices on an array, use the `symcfg list -dev` command to monitor the progress of space reclamation. The `Reclaiming` status indicates the thin device is in the process of being reclaimed and the `Total Allocated Tracks (%)` and `Total Written Tracks (%)` adjust as space is reclaimed.

Syntax
To list thin devices in an array, use the following syntax:

```
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]]
```

Options

-tdev
Lists the thin devices in the system.

-bound
Lists thin devices bound to a pool.

-unbound
Lists thin devices that are unbound.

Examples
To list the thin devices on array `341`, enter:

```
symcfg list -tdev -sid 341
```

Sample output

Note

The columns `Total Allocated Tracks (%)` and `Total Written Tracks (%)` display the percentage of the entire thin device allocated and written.

```
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

Show details about thin pools on page 535 provides information on compressing and uncompressing thin devices.

List thin device tier allocations

Examples

To list tier allocations for thin devices on array `341`, enter:

```
symcfg list -tdev -sid 341 -tier
```

Sample output

Lists both technology and disk locations for the tiers.

```
'
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
```

List thin devices in a pool

Examples

To display a list of thin devices in pool `HR_THIN_R5`, enter:

```
symcfg -sid 343 list -tdev -pool HR_THIN_R5
```

Show thin BCV devices with dynamic SRDF capability

Examples

To show device `15F9` on array `397`, enter:

```
symdev show 15F9 -sid 397
```

Sample output

`Device Configuration` displays device `15F9` as `BCV+TDEV` and `Dyamic RDF Capability` as `RDF1_OR_RDF2_Capable`.

```
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.

List data allocations across multiple pools

Options

-detail

Lists thin devices and all the pools where tracks are allocated to these devices.

Examples

```
symcfg -sid 397 -devs 1620:1630 list -tdev -detail
```

Sample output

`Total Written Tracks` displays only bound pools and represents all tracks written to the thin device for any allocation in any pool.

```
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
```

Show details about thin pools

Options

-detail
Shows details about a Virtual Provisioning pool.

-all
Shows the disabled devices in the pool.

Examples

To show details about thin pool `DG1_FBA_PL` on array `086`, enter:

```
symcfg show -pool DG1_FBA_PL -thin -detail -sid 086
```

Sample output

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 space reclamation.

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, displays in `Other Thin Devices with Allocations in this Pool`.

```
. . .

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.

Verify thin and DATA device states

Description

More than one type of device state may be queried. 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.

Syntax

To verify the state of thin or DATA devices, use 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]
```

Options

-devs
: Specifies the status of single or multiple devices.

-pool
: Returns all thin or DATA devices within a pool and must be in one of the requested states to pass verification.

-i

Specifies the polling rate. The default minimum interval value used for calculating the action is 15 seconds. A message displays at each poll. If the `-i` option is used without the `-c` option, the command loops infinitely, exiting only when all devices are in the correct state (or when manually interrupted).

-c

Specifies the numbers of polls to be executed. The command terminates at the end of the count or when all devices are in the correct state.

Verify device usage

Examples

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
```

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
```

Migrated devices

Use the `symdev list` and `symdev show` commands to provide information about devices that are changed or virtualized during a migration.

List changed devices after migration

Options

-identity_set

Displays the devices whose effective identity changed after a migrate operation.

Examples

To display the devices with identity change on array `207`, enter:

```
symdev -sid 207 -identity_set list
```

Sample output

```
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
```

List virtualized devices

Options

-identity

Displays the effective identity information of devices that are virtualized during a migrate operation.

Examples

To list identity information of migrated device `0477` on array `207`, enter:

```
symdev -sid 207 -identity show 0477
```

Sample output

```
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:Devv             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`command. Pinning available only on array arrays running Enginuity 5875 or 5876.

Pin a device

Examples

To pin the devices in group `prod1`, enter:

```
symdg -g prod1 pin
```

Unpin a device

Examples

To unpin the devices in group `prod1`, enter:

```
symdg -g prod1 unpin
```

List pinned devices

Examples

To list the pinned devices in array `341`, enter:

```
symdev -sid 341 -pinned list
```

Sample output

```
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
```

Filter device list device data

Description

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.

Options

-all
Lists all configured devices.

Syntax

To list device information with filter options, use the following syntax:

Note

For more information on device list filter options, refer to the *EMC Solutions Enabler SYMCLI Command Reference.*

```
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>]
```

Filter device list by director

-SA *#*
Lists devices by a specific front-end director number.
-p *#*

Lists devices by a specific front-end director port number.

-scsi
Lists devices mapped to SCSI front-end directors.

-fibre
Lists devices mapped to Fibre front-end directors.

Filter device list by director port mapping

-multiport
Lists devices mapped to more than one front-end director port.

-noport
Lists devices not mapped to any front-end director port.

-firstport
Lists first port information for devices with more than one port mapping schemas.

If none of these options are specified, then devices with all director-port relationships return.

Filter device list by metadevice

-meta
Lists meta member devices.

-nonmember
Lists only meta head devices.

Filter device list by SRDF devices

-R1
Lists all SRDF R1 devices.

-R2
Lists all SRDF R2 devices.

-R21
Lists all SRDF R21 devices. The -R21 option is not currently supported in conjunction with the `-dynamic` option.

-rdfg
Lists only devices belonging to the specified SRDF group number (`RdfGrpNum`).

-rdfa
Lists all SRDF/Asynchronous capable devices.

-half_pair
List all devices that are not paired with an SRDF device. Existing half pair devices can result from an SRDF/Star failover scenario, a half_deletepair operation, or a configuration change.

-dup_pair
Lists all devices that are paired with the same SRDF type. Existing duplicate pair devices can result from a SRDF/Star failover scenario or a configuration change.

-notrdf
Lists all dynamic capable devices that are not SRDF devices. Use this option to identify non-SRDF devices that have dynamic SRDF capability. In addtion, use this option with the -R1, -R2, and -R21 device options to include non-RDF devices in the list along with these specified devices.

Note

Refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide* for detailed information on SRDF device types and failover configurations.

Filter device list by FAST (Enginuity 5876 or lower)

-fast

Lists 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 a device is associated with. For example, for a thick device under a storage group that is associated with a FAST VP policy, that device is listed as part of the output results. Unbound and pinned devices are also returned as part of this filter.

Filter device list by RAID information

-raids

Lists RAID-S devices by by RAID group number. The `RAID` option lists the same information. When viewing device output for RAID-5 devices, the total device capacity (`Device Capacity`) is displaying the total logical capacity, and the hyper capacity (`Hyper Num Cap`) is displaying, for each member, the physical capacity which includes all device space—available and unavailable.

Filter device list by technology type (Enginuity 5874 or higher)

-technology

Lists the drive technology (FC, EFD, and SATA) of the primary local back-end storage of the device.

Filtering device list by DA, interface, disk, or hypervolume

-da, -dx, -interface, -disk, -disk_group, -hyper

Lists specified DAs, DXs, interfaces, disks, disk groups or hypers. These options default to `ALL` if not specified.

Filtering by storage group

-sg

Lists devices by specified storage group. This option defaults to `ALL` if not specified.

Filter device list by DIF1 attribute

-dif1

Lists device DIF1 attribute status. 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 is set on a device, it displays as `TRUE` or `FALSE`.

Restrictions

- DIF1 attribute is 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.
- Devices must be unmapped while resetting the DIF1 attribute.
- Devices 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, 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 fails.
- 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 relationship between the DIF1 attribute and replication. Both source and target devices of any replication can have their own DIF1 setting.

Filter device list by AS400_GK attribute

-as400_gk

Lists device AS400_GK attribute status. The AS400_GK attribute setting on an array device is required when a AS400 device is used with IBM host control software. The attribute is also used with the Celerra Network Attached Storage for Celerra gatekeeper devices. If the AS400_GK attribute is set on a device, it displays as `TRUE` or `FALSE`.

Refer to esd_r_se_seam_setting_device_attributes_687650 on page 265 for detailed information on setting the device attributes.

Restrictions

- AS400_GK attribute is 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 the AS400_GK attribute is 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 with 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 list capacity output options

-cyl
Lilsts device capacity in cylinders rather than the default of megabytes (MB).

-pd
Lists only the host visible devices (pdevs).

Device external locks

Solutions Enabler uses *device external locks* in the array to lock pairs during replication operations (such as Open Replicator, TimeFinder, and SRDF operations).

NOTICE

Use the release lock action only if it is determined 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 held by certain applications (such as an SRDF action) are longer duration locks.

List devices with external lock

Examples
To list a range of devices (`0000:000A`) that have a device external lock, enter:

```
symdev list -sid 870 -devs 0000:000A -lock
```

Release devices with external lock

Examples
To release the locks held by the configuration server for device `204` on array `343`, enter:

```
symdev release -sid 343 devs 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.
```

Note

Aborting the session, if successful, releases all locks held by the session including device locks.

Disk-level data

The configuration database file maintains low-level configuration and status information for each disk on every array that are accessible from the host, including external spindles (eDisks). Use the `symdisk list` and `symdisk show` commands, to list all the available disks, or specify individual disks to obtain configuration and status information.

List and show disk information (HYPERMAX OS 5977)

Syntax

To display disk details, use the following syntax:

```
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]
```

Options

-engine
Lists attributes, counts and capacities relevant to Internal Disk Group/Engine Spindle Groups.

-cyl | -mb | -gb | -tb
Indicates how to report disk capacity.

List and Show Disk Information (Enginuity 5773 through 5876)

Syntax

To display disk details, 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.

Examples

To list details for disk `07a:c0` on array `432`:

```
symdisk show 07a:c0 -sid 432
```

Sample output

```
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
  }
```

Note

When virtual devices (VDEVs) are part of the disk configuration, the capacity shown for the virtual hypers reflects only space to store track tables (pointers). Typically, a low number, such as 3 MB, is needed for these devices.

List SAS drives

Description

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).

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.

Options

-v
Lists the SAS technology type in the `Technology` field. The `Form Factor` field displays 2.5, 3.5, or N/A to indicate the Enterprise Flash Drive.

List disk gaps

Options

-gaps
Used with the `symdisk list` command, and lists any gaps found on the disk.

-spare_info
Used with the `symdisk list` command, and shows which disk the spare disk is substituting for, if it is invoked.

-gaps_only
Used with the `symdisk show` command, and lists any gaps found on the disk without listing all of the hyper information.

Reported actual disk capacity vs. rated disk capacity

The `Actual Disk Capacity` lists 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 for arrays running Enginuity 5875 or higher.

List disk groups

Options
Used with the `symdisk list` command.

-by_diskgroup
Lists the disks by disk group number.

-disk_group
Limits disks that belong 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.

-dskgrp_summary
This option displays summary information for disk groups. Without the `-v` option the disk group summary information is displayed in a table format. With the `-v` option the disk group information in expanded format (one field per line) is displayed, information regarding spare disks is not included.

Note

The `-dskgrp_summary` option is not valid with the `-hypers`, `-spare_info`, `-gaps`, `-isspare` and `-failed` options.

List disk group summary (HYPERMAX OS 5977)

Examples

To list disk group summary for array `584`, enter:

```
symdisk list -dskgrp_summary -sid 584
```

Note

The command line accepts `-dskgrp_summary` abbreviated as `dskg`.

Sample output

```
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
```

Output field changes for arrays running HYPERMAX OS 5977:

- `Hyper Size` — Added column. Lists the hyper size used for each internal disk group/engine spindle group. For arrays running Enginuity 5876 or lower, the hyper size for external (FTS) disk groups or internal disk groups reports as `ANY` as these groups do not use a consistent hyper size.
- `Free Capacity` — Reports as `0` as the Spindles in these Disk Groups are fully provisioned.
- `Actual Capacity` — Removed column.

Output with `symdisk list` verbose (`-v`) option:

```
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
.....
```

Verbose output field changes for arrays running HYPERMAX OS 5977:

- `Actual Group Capacity` — Removed field.
- `Summary Actual Group Capacity` — Removed field.
- `Max Hypers Per Disk` — Added field. Listed for each disk group.
- `Hyper Size` — Added field. Listed for each disk group.

List disk groups by engine (HYPERMAX OS 5977)

Options

-by_engine
Lists attributes, counts and capacities specific to internal disk group/engine spindle groups.

-detail
Expands the list to include the disk group name.

Examples

To list disk groups by engine for array `584`, enter:

```
symdisk list -dskgrp_summary -by_engine -sid 584
```

Note

The command line accepts`-dskgrp_summary` abbreviated as `dskg`, and `-by_engine` abbreviated as `-by_eng`.

Sample output

```
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
```

Output field changes for arrays running HYPERMAX OS 5977:

- `Usable 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 internal disk group/engine spindle group) for internal disk groups. For arrays running Enginuity 5876 or lower, external (FTS) disk groups or internal disk groups report the `Disk` and percent (`%`) columns as `N/A`.
- `Total Spare Coverage` — Reports in units of `Disk` and percent (`%` for internal disk groups/engine spindle groups. The Disk-based units correspond to the `Disk Size` and `Max Hypers Per Disk` reported in the `-dskgrp_summary -v` output. 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. For arrays running Enginuity 5876 or lower, external (FTS) disk groups or internal disk groups report the `Disk` and percent (`%`) columns as `N/A`.
- `Available Spare Coverage` —Reports in units of `Disk` and percent (`%` for internal disk groups/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 `Available Spare Coverage` values are the amount of spare capacity currently available for sparing and this percent is relative to the `Total Spare Coverage` value. For arrays running Enginuity 5876 or lower, external (FTS) disk groups or internal disk groups report the `Disk` and percent (`%`) columns as `N/A`.

Output with `symdisk list (-detail)` option:

```
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
```

List spare physical disks

Description

Dynamic sparing is only supported on arrays running Enginuity 5873 or lower. Permanent sparing is supported on arrays running Enginuity versions 5874 or higher.

Options

Use the following options with `-DA`, `-interface`, or `-tid` options to list spare disk information on specific directors, disk interfaces, or SCSI target IDs .

-isspare

Lists information about spare physical disks (spindles), if a spare disk is invoked against a failed disk.

-v -spare_info

Lists information about the failed disk that has been replaced. The `-v` option must be specified with this option. This display option is not applicable for permanent sparing.

-gaps

Lists gaps found between hypers on the disk, as the hypers are listed. To see a short list of only the gap information, 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 approximate values. 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.

-nospares
Lists only internal disks (spindles) that are capable of being covered by a spare, but currently are not. This option is only supported on Enginuity version 5876 and higher. No disks (spindles) are displayed if this option is issued for an array running Enginuity version lower thant 5876. This option is not valid for external spindles (external spares do not exist) and no external spindles display when this option is specified.

Examples
To list information spare disk information about failed disk, enter:

```
symdisk list -v -spare_info
```

Sample output
Lists a spare disk that is invoked against disk `16B:C0`

```
...
  Spare Disk                : False
  Director                  : DF-16B
  Interface                 : C
  Target ID                 : A
  Disk Group Number         : 0
  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
```

List spindle information

Options

-spindle
Lists each disk by Spindle ID and includes the director number for each spindle.

-spid *Spindle_ID*
Lists physical disks by Spindle ID.

Note

For Enginuity versions 5875 or lower, the specified Spindle ID does not represent an actual Enginuity entity, but is a string representation of a unique hexadecimal number. This number can be used as an input parameter for `symdisk show` commands.

Examples

To list the spindle IDs for array `306`, enter:

```
symdisk list -spindle -sid 306
```

Sample output

```
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
```

List spindle information for physical disk

Syntax

-spid *Spindle_ID*
Specifies spindle ID.

Examples

To list spindle information for spindle `1B935` on array `306`, enter:

```
symdisk show -spid 1B935 -sid 306
```

Note

The `sympd list` command with the `-spindle` option also provides spindle ID information. List array device information on page 522 provides more information.

Sample output

Note

This is the same output format as `symdisk list -v` (verbose).

```
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 identifies remote Data Domain appliances when reporting on virtualized disks and LUNs on external arrays. If applicable, DataDomain displays in the `Product ID` field.

List spindle information for disk group

Options

-spindle
Lists each disk by Spindle ID and includes the director number for each spindle.

Examples

To list spindle IDs for disk group `004` for array `306`, enter:

```
symdisk list -disk_group 004 -spindle -sid 306
```

Sample output

```
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
```

External spindle information

Federated Tiered Storage attaches external disks (eDisks) to a VMAX arrays, and makes the eDisk's capacity available to the VMAX array as an external spindle.

Use the `symdisk show` command to list the spindle IDs for eDisks.

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, and external provisioning for XtremIO edisks. Refer to Federated Tiered Storage on page 361 for more information on FTS.

View external spindle information

Syntax

To list external spindle information, use the following syntax:

```
symdisk show -wwn wwn
```

Output fields for external spindle information

- `Disk Location` — Shows `Internal` for internal disks, and `External` for external disks. For Enginuity 5876 or lower, `Disk Location` shows as `Internal`.
- `Disk Service State` — Shows the availability of the eDisk. 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.
- `Technology` — Shows as `N/A` for external disks.
- Shows if an eDisk is encapsulated or not.
- Shows the number of external paths, available to the external LUN, to reach the eDisk.

CHAPTER 16

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.

- Statistics overview 556
- REQUESTS statistics 557
- BACKEND statistics and PREFETCH activity 559
- Director statistics 560
- PORT statistics 561
- List ISCSI statistics (GigE) 561
- List MEMIO statistics 562
- Disk statistics 563
- CACHE statistics 564
- List PATH statistics 566
- List CYCLE statistics 566
- RDF statistics 568

Statistics 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.

Object reporting statistics

The SYMCLI `symstat` command returns statistics on the how the following objects are performing in the storage 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.

Display options for performance statistics

Options

The `symstat` command provides the following performance display options:

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.

REQUESTS statistics

The `symstat` command with the *REQUESTS* type displays performance 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 theesd_r_se_seam_setting_compatibility_mode_for_older_solutions_enabler_versions_404421 on page 33 for information on returning output as seen in prior Solutions Enabler versions.

List array statistics

Examples

To return performance statistics for array `345` at 5-second intervals, enter:

```
symstat -i 5  -type REQUESTS -sid 345
```

Sample output

```
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
```

List device statistics

Examples

To list device throughput statistics at 120-second intervals for three counts on device `DEV001` , enter:

Note

`Requests` type is the default.

```
symstat -i 120 -c 3 -g prod_r1 -ld DEV001
```

List SRDF/A session statistics

Syntax

To return SRDF throughput information for SRDF/A sessions, use the following syntax:

```
symstat -repType rdfa
```

When requesting SRDF/A session statistics for a specified array, the output differs depending upon the additional qualifiers used with the `-sid` flag arguments specified:

- If used with no further qualifiers, then the output display lists SRDF/A sessions that have a status of *active*.
- If used with qualifier `-g all`, then the output display list all device groups, regardless of status.
- If used with the qualifier `-rdfg ALL`, then the output display lists all SRDF groups, regardless of status.
- If 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 (`-sid` option is not used), then the output display only lists details for the specified device group.
- If used with the qualifier `-rdfg #`, then the output display lists just the specified SRDF group.
- If used with the qualifier `-cg` *CgName*, then the output display lists just the specified composite group.

- If used with the `-v` option the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

List R-Copy session statistics

Syntax
To return statistical data for an rcopy session, use the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name SessionName
```

Examples
To return R-Copy session statistics at 60-second intervals for three counts for array `0003`, enter:

```
symstat -sid 0003 -reptype rcopy -i 60 -c 3
```

BACKEND statistics and PREFETCH activity

The `symstat` command returns statistics for I/O requests on back-end directors, the throughput on the corresponding devices, and prefetch disk activity on back-end directors.

List BACKEND statistics

Description
Use the `symstat` command with the `BACKEND` type specified, to 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 to examine for I/O requests and throughput activity.

Examples
To list statistical data every 60- econds for 3 counts for logical device `DEV001`, enter:

```
symstat -type BACKEND -i 60 -c 3 -g prod_r1 -ld DEV001
```

List PREFETCH activity

Examples
To list prefetch disk activity on back-end directors, at 60-second intervals for four counts, enter:

```
symstat -i 60 -c 4 -sid 333 -da all -type prefetch
```

Solutions Enabler with Enginuity 5876 or HYPERMAX OS 5977 provides the `-dx` option for the `symstat` command which allows prefetch disk activity on external directors (DXs).

Examples
To list prefetch disk activity on `-dx` directors at 60-second intervals for four counts, enter:

```
symstat -i 60 -c 4 -sid 333 -dx all -type prefetch
```

Director statistics

The `symstat` command displays the I/O requests and throughput activity performance statistics on any or all directors. In addition, it returns director statistics for `rcopy` sessions. To return a list of directors available on all visible arrays, use the `symcfg list -dir all` command.

Note

For arrays running Enginuity 6874 or higher, use the `-dir -EF` option to run statistics for FICON directors .

List director I/O requests

Examples

To output statistics for all directors at 60-second intervals for 100 counts 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 director statistics

Syntax

To return rcopy statistical data for a specific session, use the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name SessionName
```

To return `rcopy` session statistics at 60-second intervals 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

The `symstat` command with the `PORT` type specified, displays performance statistics such as I/O requests and throughput of director ports for a specific array.

List director port statistics

Syntax

To report director port statistics, use the following syntax:

```
symstat [-i Interval] [-c Count] -type PORT [-sid SymmID]
   -dir <#|ALL> [-port <#|ALL>]
```

Examples

To output director port statistics in 60-second intervals for 3 counts on array `0256`, enter:

```
symstat -i 60 -c 5 -type PORT -sid 0256 -dir ALL
```

List rcopy session director port statistics

Syntax

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.

List ISCSI statistics (GigE)

Description

The `symstat`command with the `iSCSI` type specified, displays 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.

Options

--tcp_conn

Returns TCP connection-based statistics.

--gige_dir
Returns network interface-wide statistics.

-rdf_spdlmt
Returns SRDF speedlimit statistics.

--gige_dir
Returns SRDF network compression statistics.

-spdlmt_grp
Returns SRDF speedlimit group statistics

Examples

Note

For all of the examples below, the SRDF director number must be specified. (`ALL` is a valid value for the director.)

To return TCP connection-based statistics, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -tcp_conn -DIR 20
```

To return network interface-wide statistics, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -gige_dir -DIR 20
```

To return SRDF speedlimit statistics, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -rdf_spdlmt -RE 24
```

To return SRDF network compression statistics, 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, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -rdf_spdlmt -RE 24
```

List MEMIO statistics

Description
The `symstat` command with `MEMIO` type, displays performance statistics for cache memory to disk I/O on a device group (use `symdg list`command to list the current device groups).

Options

-bcv
Returns statistics for only BCV devices in the device group.

-all
Return statistics for both standard and BCV devices in the device group.

Examples
To report cache memory to disk I/O in 5-second intervals on device `0383`, enter:

```
symstat -i 5 -g Prod -ld DEV001 0383 -type MEMIO -sid 345
```

Sample output

Note

By default, BCV devices are not included in these results.

```
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
```

Disk statistics

The `symstat` command, with the `DISK` type specified, displays performance statistics such as I/O requests and throughput on a physical disk.

List statistics for a specific disk

Examples

Note

To retrieve the statistics of all DA directors, interfaces, or disk SCSI IDs, use the value `ALL`.

To report statistics in 60-second intervals for 3 counts on disk **`02A:C5`**, enter:

```
symstat -type disk -i 60 -c 3 -sid 577 -disk 2a,C,5
```

To report statistics at the same interval and count, but for all SCSI IDs on disk **`02A: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.

List spindle statistics

Description

For arrays running Enginuity 5876 and higher the `symstat` command with the `-spindle` option requests statistics by Director and Spindle ID for internal or external (DX director) disks.

Syntax

To list spindle statistics, use 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>]]
```

Options

-spid

Requests statistics for a disk by specifiying a Spindle ID, or for `ALL` disks.

Examples
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

The `symstat` command, with the `CACHE` type specified, returns data about the cache activity on the front-end directors, LRU, or SRDF/A sessions.

Note

LRU statistics are not available for arrays running HYPERMAX OS 5977 or higher.

List front-end director cache statistics

Description
The `symstat` command returns statistics for front-end directors with either specifying (`ALL`) directors or a specific director type (`SA, EA, FA, FE, CA, RA`).

Example
To report cache activity in 60-second intervals for 4 counts for all SA directors on array `097`, enter:

```
symstat -i 60 -c 4 -sid 097 -sa all -type cache
```

List SRDF/A session cache statistics

Description
The `symstat` command returns 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 requesting SRDF/A session statistics for a specified array, the output differs depending upon the additional qualifiers used with the `-sid` flag arguments specified:

- If used with no further qualifiers, then the output display lists SRDF/A sessions that have a status of *active*.
- If used with qualifier `-g all`, then the output display list all device groups, regardless of status.
- If used with the qualifier `-rdfg ALL`, then the output display lists all SRDF groups, regardless of status.
- If 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 (`-sid` option is not used), then the output display only lists details for the specified device group.
- If used with the qualifier `-rdfg #`, then the output display lists just the specified SRDF group.
- If used with the qualifier `-cg CgName`, then the output display lists just the specified composite group.
- If used with the `-v` option the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Examples

To return SRDF/A session cache 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. 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 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. 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 results 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. A cache slot with both a local write pending and an SRDF/A write pending will 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.

List PATH statistics

Description
Path statistics for non-incremental `rcopy` sessions can be reported. Arrays that have all or some incremental sessions will report an error.

The reported data can either be listed by session, where the data is organized by session name (`-by_session`) or by port (`-by_port`) where data will be organized by director port

Examples
To report the `rcopy` path statistics for array `003` and organized by director port, enter:

```
symstat -sid 0003 -reptype rcopy -type path -by_port -i 60 -c 3
```

Sample output

```
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
```

List CYCLE statistics

Description
The `symstat` command with the `CYCLE` type statistics returns a table of cycle information for device groups (SRDF/A sessions) in arrays that have active status or have nonzero cache usage.

When requesting SRDF/A session statistics for a specified array, the output differs depending upon the additional qualifiers used with the `-sid` flag arguments specified:

- If used with no further qualifiers, then the output display lists SRDF/A sessions that have a status of *active*.
- If used with qualifier `-g all`, then the output display list all device groups, regardless of status.
- If used with the qualifier `-rdfg ALL`, then the output display lists all SRDF groups, regardless of status.
- If the `-g` *DgName* device group qualifier is used (`-sid` option is not used), then the output display only lists details for the specified device group.

- If used with the qualifier `-rdfg #`, then the output display lists just the specified SRDF group.
- If used with the `-v` option the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Examples

To return SRDF/A session cycle statistics of active sessions for array `0011` , enter:

```
symstat -type cycle -i 60 -c 1 -sid 0011 -RepType rdfa -g all
```

Sample output

```
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 remote arrays. These statistics are used to monitor on-going activity and detect problem areas.

Note

RDF directors are supported on VMAX 10K, 20K, and 40K arrays running Enginuity 5876. A `symstat` command specifying a RDF director using the `-dir` option produces the appropriate output if RDF directors exist on these arrays. The `symstat` command produces the appropriate output when issued against these arrays with RDF specific options. These options include `-hop2`, `-type RDF`, `-RepType rdfa`, `-RepType rdf`, and `-type REQUESTS -RA`.

List SRDF/A session statistics by RA group

Options

all
Returns statistics for all RDF groups.

Examples

To return statistics for all RDF group `-rdfg 11`, enter:

```
symstat -sid 179 -i 60 -c 2 -type RDF -rdfg 11
```

Sample output

```
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
```

List SRDF device-level statistics

Description

Device-level statistics are 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 also be specified when a logical device name (`-ld LDevname`) is used.

Example

To return device-level statistics for device `1085`, enter:

```
symstat -i 5 -sid 345 -dev 1085 -type RDF
```

Sample output

To return SRDF device-level statistics for device `1085`, enter:

```
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
```

List SRDF statistics by director number

Options

all

Returns statistics for all RDF groups.

Example

To return statistics for director `1d`, enter:

```
symstat -sid 179 -i 60 -c 2 -type RDF -dir 1d
```

Sample output

```
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)

Options

-rdflink

Returns SRDF link throughput statistics.

Example

```
symstat -sid 179 -i 60 -c 2 -type RDF -dir 2c -rdflink
```

Sample output

To return link-level details, specify the **-rdflink** option as shown in the following example.

```
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
```

CHAPTER 17

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.

- Change Tracker overview 572
- Practical uses for Change Tracker 572
- Change Tracker command summary 573
- Monitor data objects for changes 574
- Create a Change Tracker session 575
- Change tracker reports 579
- View change tracker reports 579
- Change Tracker for SRDF capacity planning 582

Change Tracker overview

NOTICE

Change Tracker is not supported on arrays running HYPERMAX OS 5977 or higher.

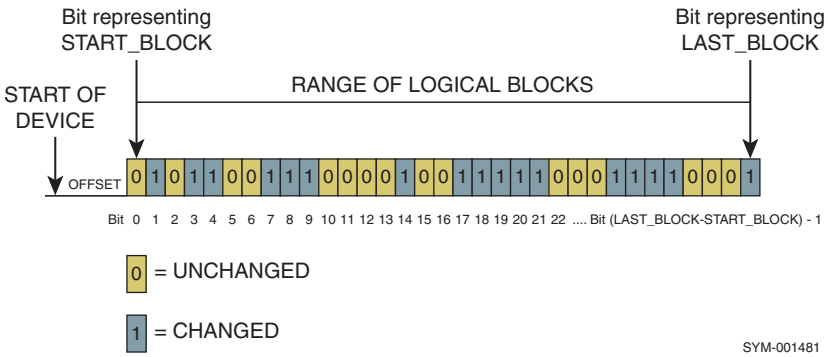
The EMC Solutions Enabler Change Tracker software measures changes to data on an array volume or a group of volumes. The Change Tracker command (`symchg`) is used on FBA devices to mark selected areas of disk storage so that the logical volume occupying those areas can be monitored for change.

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.) The following below illustrates a DeltaMark bitmap table.

Figure 13 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 for Change Tracker

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 `symchg` command monitors selected disk storage objects using DeltaMark bitmap technology to track changes and generates reports for analysis. Change Tracker monitors 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.

The following table summarizes the SYMCLI `symchg` commands used to create a Change Tracker session, mark areas to be monitored, remove or delete a session, and generate a report.

Table 55 Change Tracker command summary

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.
	`remove`	Unmarks the disk storage object while retaining the Change Tracker session. This command does not remove devices from the Change Tracker database if the object type does not match the object type in the command. If this command fails before the end of ts device list, a `remove` must be issued on each device that was not removed. In this case, use the `-force` option to force the removal of all the devices from the database. The `--force` option is used with `symchg delete` command to force the deletion of the Change Tracker session from an array and the deletion of the devices from the Change Tracker database.
	`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 Monitor data objects for changes on page 574. For details about the `symchg` command syntax and other SYMCLI commands, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Monitor 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.

Create a Change Tracker session

Syntax

Before marking objects, create a Change Tracker session for the device(s) containing the objects by using the symchg create command. To create a Change Tracker session, wse the following syntax:

```
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]
```

Set Change Tracker session type

Syntax

The `symchg create`command 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. More the one object can be marked and examined per session, and a device name or logical device name can be specified within a device group or composite group for the session. BCVs of a device group can also be specified.

Options

-type

Displays the session type. The types 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)* — Performs both a read and a write on a device.

Examples

To create a Change Tracker session of type *Write* (default) for a single device (17C) on array ID 55, enter:

```
symchg create dev 17c -sid 55
```

Note

If the session type is *Write* (default) it does not need to be specified in the command.

Create a Change Tracker device text file

The device file syntax for a Change Tracker session contains a single column of devices specified by the device name (*SymDevName*). Lines in the device file that begin with a pound symbol (#) are ignored. The device filename (`-file FileName`) is inserted into the command line for control operations.

Sample device file

```
# dev_file_1
# Symmetrix Device List for Change Tracker session
```

```
#
00A
00B
00C
057
058
# End
```

Mark a data object for change monitoring

Description

After creating a session, 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.

When an object is marked, an entry is added or updated for the specified object in the `symchg` database. The mark command resets the DeltaMark bitmaps for the devices of the specified object.

A device name or a logical device name can be specified as the marked object. BCVs of a device group can also be marked.

Syntax

To monitor a session, use the following syntax:

```
symchg [-sid SymmID] [-v]
    mark   dev SymDevNamesymchg -g DgName | -cg CgName [-bcv] [-v]
    mark   [ld LdevName]
symchg -sg SgName -sid SymmID [-v]
    mark
symchg -file DevFile -sid SymmID [-v]
    mark
```

Note

Use the same convention for the `mark` command as was used with the `symchg create` command. If the session was created using a device group, composite group, storage group, or device file, the device marking must be managed using the same method.

Examples

To mark device (`17C`) on array ID `000125600055`, enter:

```
symchg mark dev 17c -sid 55
```

List objects marked for change monitoring

Examples

Use the `symchg list` command to list disk storage objects that are 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 -type r
```

Remove mark from change monitoring database (without termination)

Description

A storage object can be unmarked from the `symchg` database without terminating the Change Tracker session. The `remove` command causes the DeltaMark bitmaps to stop monitoring changes for the devices, while maintaining the session.

A device name or a logical device name can be specified for unmarking. BCVs of a device group can also be unmarked.

Syntax

To remove a disk storage object from the without terminating the Change Tracker session, use the following 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

Use the same convention for the `remove` command as was used with the `symchg create` command. If the session was created using a device group, composite group, storage group, or device file, the device marking must be managed using the same method.

Examples

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
```

Remove marked objects from change monitoring database (with termination)

Description

A storage object can be unmarked from the `symchg` database and delete (terminate) the Change Tracker session. The `delete` command causes the DeltaMark bitmaps to stop monitoring changes for the devices and deletes the session.

A device name or a logical device name can be specified for unmarking. BCVs of a device group can also be unmarked.

Syntax

To unmark a storage object and terminate the Change Tracker session,, use the following 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

Use the same convention for the `delete` command as was used with the `symchg create` command. If the session was created using a device group, composite group, storage group, or device file, the device marking must be managed using the same method.

Examples

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` and delete the session, enter:

```
symchg -g ProdDB delete -bcv
```

To unmark devices listed in the device text file `dev_file_1` and delete the session, enter:

```
symchg -file dev_file_1 delete
```

View objects marked for change monitoring

Description

The DeltaMark bitmap can be processed to display the amount and the rate of change that corresponds to a storage object in the Change Tracker database.

Options

-session
: Verifies the existence of a Change Tracker session for a specified object by including the **-session** option.

-g and -bcv
: Displays the amount and the rate of change for BCV devices in the device group. **ProdDB**, enter:

-native
: Returns output in the native track size, which normally defaults to 32K track size when saving marked objects to a log file.

-reset
: 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.

-multi

Sets the changed tracks for collection in both the *SUM* and *DELTA* collection methods.The SUM collection method calculates the cumulative number of changed tracks from the starting point of when the object was first marked.

Examples

To display the amount and the rate of change for device (`17C`) on array ID `000125600055`, enter:

```
symchg view dev 17c -sid 55
```

To display the amount and rate of change for BCV devices in device group `ProdDB`, enter:

```
symchg -g ProdDB view -bcv
```

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
```

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
```

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.

Change tracker reports

Change Tracker generates 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.

View change tracker reports

Syntax

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
```

Options

-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.

-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.

-backend
Specifies a storage group on which to report.

-sg
Displays information about the DA and back-end disk address of marked devices.

-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: Creating a report of change tracking information

The following is an example of typical steps to follow when saving change information to a user-defined log file and generating a report:

Procedure

1. Create Change Tracker sessions for each standard device in a device group (DATABASE_DG).

```
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.

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 above 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).

```
symchg report -g DATABASE_DG -log /tmp/change_data.log -v
```

The data in the log file is in a comma-separated format, so it can also import it into a spreadsheet to view the data.

5. 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.

```
symchg report -g DATABASE_DG -log /tmp/change_data.log -rate 1000 -
ra 2
```

6. View information for a device group (DATABASE_DG) on your screen showing the log entries.

```
symchg view -g DATABASE_DG
```

7. List the objects that are marked and being monitored for change.

```
symchg list
```

8. Verify a Change Tracker session has been created for a device group (DATABASE_DG) and that it is marked.

```
symchg view -g DATABASE_DG -session
```

9. Remove the marked object (DATABASE_DG).

```
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).

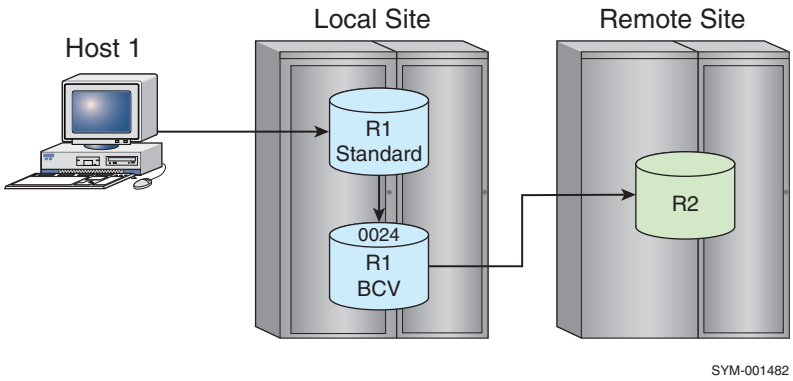
Change Tracker for SRDF capacity planning

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 the following figure, 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 14 SRDF configuration



Using Change Tracker to perform SRDF capacity planning

The following steps outline an example of setting up devices and using Change Tracker to perform SRDF capacity planning:

Procedure

1. Identify the device group to monitor. If the group does not exist, create the device group `deltadg`.

   ```
   symdg create deltadg
   ```

2. Add a device 085 to the device group.

   ```
   symdg -g deltadg -sid 3264 add dev 085
   ```

   By default, SYMCLI assigns a logical device name (in 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 are measured).

   ```
   symchg -g deltadg mark
   ```

6. In a log file, record cumulative track changes that occur on each device in the device group. Collect 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 18

Optimizing Array Performance (Enginuity 5876 or lower)

This chapter describes Optimizer concepts and how to improve array performance using the SYMCLI **symoptmz** command.

- Optimizer overview 586
- Optimizer operations 588
- Optimizer operations using the symoptmz command 592

Optimizer overview

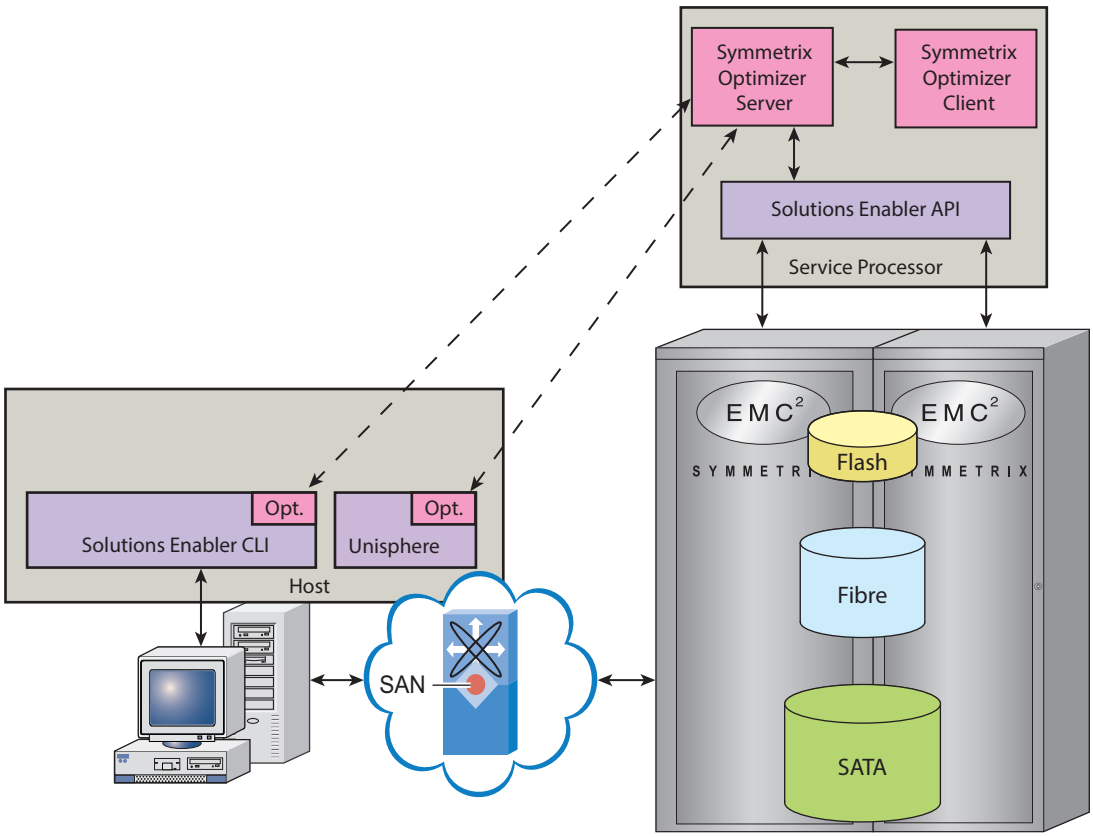
Note

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 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 15 on page 586.

Figure 15 Optimizer application architecture



Optimizer works within a single array and does not change the protection of a device.

From a host system, Solutions Enabler CLI commands and Unisphere dialogs communicate with the Optimizer server. The SYMCLI **`symoptmz`** command provides all the necessary options to perform Optimizer operations.

Note

The Optimizer client interface is used by EMC Customer Service only.

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 device swaps.

The second part of the chapter describes the **symoptmz** command and explains the options used to manage Optimizer on the array.

Optimizer configuration requirements

Optimizer can optimize all open system and mainframe RAID 1 (mirrored), RAID 5, RAID 6, and RDF-protected devices in an array.

Restrictions for device swap
To be considered for a swap, devices must be:

- Same size and emulation
- Same protection (Enginuity 5874)
- Available DRVs (Dynamic Reallocation Volumes) for both source and target
- Currently attached to legal RAID groups

Unsupported devices
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 devices
- DATA devices
- DRV devices
- Devices marked by user as "Do not move"
- SFS devices
- Striped CKD devices
- Thin devices
- Vault devices
- Virtual / SAVE devices

Optimizer configuration requirements (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.

Optimizer configuration requirements (Enginuity 5773 or lower)

DRV devices
At least two DRV devices 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, two 4 GB open system DRV devices must be configured. If more than one simultaneous swap is desired, then additional DRV devices are needed (two for each swap). If there are volumes of mixed size or emulation within the same array, then DRV devices are required for each size and type to be swapped.

Open mirror position
Optimizer requires an open mirror position to perform a swap; however, arrays running Enginuity 5773 or lower have a four mirror slot limitation per volume. This limitation may make it difficult for Optimizer to obtain the required mirror position to perform a swap. For example, if 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, it will drop the swap and rerun the analysis. If Optimizer is in Rollback, Manual, or User-Approved mode, then Optimizer retries a number of times until an error is encountered. Eventually, Optimizer stops trying if a mirror slot does not become available.

Configuration lock requirement
For arrays running Enginuity 5773 or lower, Optimizer holds a Symmetrix External Lock (SEL) to perform a swap or migration. The configuration lock is an exclusive lock on the array that 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 behaves as previously described (no open mirror position). Optimizer holds the configuration lock for the duration of the swap or migration, or until the operation is canceled, so other applications cannot 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.

Optimizer operations

After it has been initialized with the user-defined parameters, Optimizer operates autonomously on the array service processor. Optimizer monitors and controls array performance as follows:

1. Optimizer builds a database of device activity statistics on the array back-end. This data is saved for a maximum of two weeks.
2. Using the statistical data collected, configuration information, and 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 evaluates the speed of the disk, the disk geometry, and the actuator speed to determine faster disks.
3. The array device swaps are performed. User-defined parameters control whether swaps occur in a completely automated fashion or require approval before the action is taken.
4. Optimizer continues data analysis for the next swap.

Optimizer 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.

Optimizer 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 seek, latency and transfer time.

Seek time
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
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
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.

Optimizer device swap determination

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 the following strategies when determining which swaps to make:

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 contention on the individual physical disks, improving device performance regardless of activity level.

Minimizing seek
Performs swaps that relocate highly active devices so that they are closer together on the physical disk and thus decrease the seek distance for I/Os to these devices. When Optimizer swaps devices, it tries to achieve this configuration to decrease overall I/O service time.

Using faster media
Optimizer tries to swap highly active devices to 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 gig-to-gig seek times, zone-bit recording, and bandwidth data.

Optimizer swap determination

The Optimizer analysis consists of three high-level phases:

Calculate service time
Model and sum the total service time of each disk 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 to guarantee that the best swap is selected.

The maximum number of hypers per physical disk is automatically set to 512 for Enginuity 5874 and 1024 for Enginuity 5876. For Enginuity 5773 and lower, 1 to 256 hypers are allowed per disk. Potential swaps are assessed by *M0* rating, which 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, Optimizer also ensures that the new maximum of service times is less than the old one.

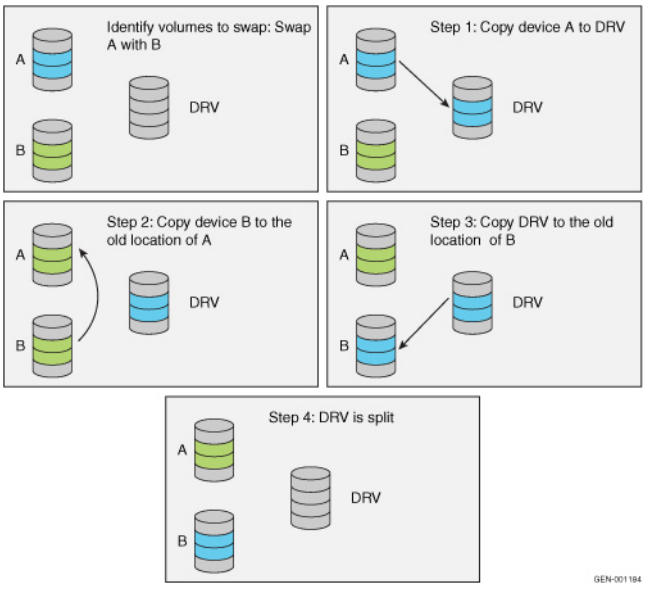
Optimizer swap procedural overview (Enginuity 5874 to 5876)

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 Multi-session device creation (Enginuity 5773 through 5876) on page 240 for information about creating mirrored DRV devices.

Hypervolumes are swapped using a four-step process. One DRV must be configured per swap. The DRV should be of equal or greater size than the swapped devices. With Optimizer, up to eight simultaneous swaps can happen at once.

Figure 16 on page 590 illustrates the steps for swapping device A with device B.

Figure 16 Swap steps using one DRV



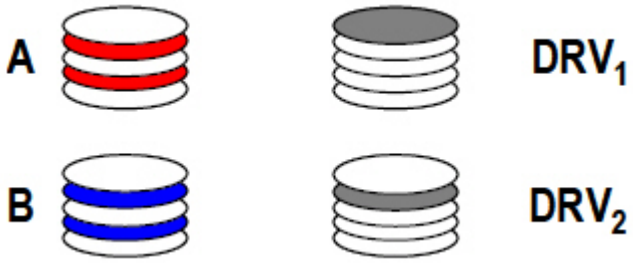
Optimizer swap procedural overview (Enginuity 5773 or lower)

The swap procedure relies on TimeFinder technology and uses DRVs as temporary mirrors. The procedure describes the swap steps when dealing with mirrored device 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 17 on page 591, assume the red volumes (on A) have high activity and the blue volumes (on B) have low activity.

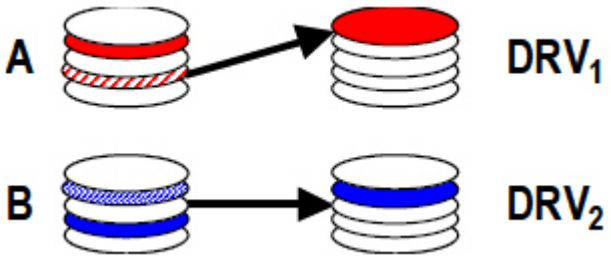
Figure 17 Identify volumes to swap



Step 2: Copy volume to DRV

Optimizer swap commands are passed to SymmWin, which assigns DRV1 as a third mirror for hypervolume A and DRV2 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 18 on page 591). Both hypervolumes still have two (or more) physical mirrors. Host activity to the hypervolumes is now directed to DRVs and the other mirror.

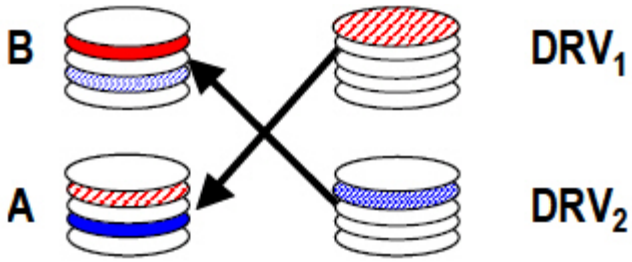
Figure 18 Copy volume to DRV



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 19 on page 592). This is similar to a BCV restore.

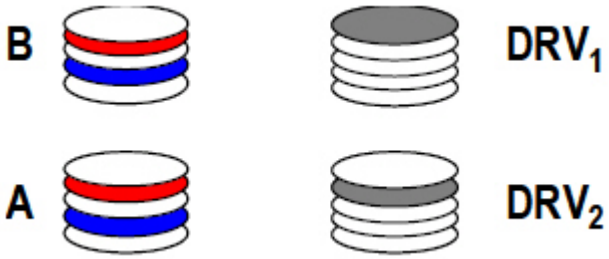
Figure 19 Copy DRVs to new locations



Step 4: DRVs are split

The final step is to split the DRVs from their standard and mirror hypervolumes after synchronization completes (see Figure 20 on page 592). The drive's balance improves and the DRVs are now available for the next swap.

Figure 20 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 overall array performance, Optimizer will not include the same spindle twice in one swap group. QoS controls can be used to set a lower BCV priority for the STDs to be swapped. Managing Quality of Service on page 451 explains how to use QoS controls. The Time Window feature can be used to plan the swap activities to occur on idle business time. Managing Time Windows on page 605 explains how to use the Time Window feature.

Optimizer operations using the symoptmz command

The SYMCLI `symoptmz` command provides all the necessary actions to control Optimizer behavior. Some command actions are executed directly on the command line, while others are processed in a command file. The `symoptmz` command provides the following actions:

clear_stats
: Clears disk statistics maintained by Optimizer.

commit
: Updates Optimizer with the changes defined in the command file.

disable
: Disables Optimizer algorithm processing.

enable

Enables 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 its current state and version information. 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.

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**.

Enable and disable Optimizer

Description

When enabled, Optimizer collects data and performs data movement. When disabled, Optimizer does not perform data movement.

Syntax

To start or stop the Optimizer process on the array service processor, use the following syntax:

```
symoptmz enable|disable
```

Commit command file changes

Description

Place commands for setting or clearing any parameters or time windows in a command file, which is processed by `symoptmz`. Alternatively, use `stdin` redirection with "here documents" in UNIX shell scripts.

Terminate each command in the file with 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 of the commands in a 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 Optimizer parameters, use a command file with the `commit` option. The `commit` option performs syntax and range checks, and then updates Optimizer with the modified parameters.

Example

To commit the changes specified in the command file `opt_config.txt`, issue the command:

```
symoptmz -file opt_config.txt commit
```

Clear Optimizer statistics

Syntax

To clear the disk statistics maintained by Optimizer, use the following syntax:

```
symoptmz -sid SymmID clear_stats
```

Set control parameters

Description

Control parameters can be set for Optimizer or FAST. Both tools use the same command file format.

Syntax

For the command file, Use the following syntax:

```
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];
```

Options

start_mode
Determines whether Optimizer is enabled or disabled when Optimizer is launched (for example, after rebooting the service processor).

swap_mode
Controls how Optimizer swaps devices:

- `AUTO`
  Automatically swaps devices as soon as it finds swaps that would improve performance. Each day, Optimizer performs up to the number of swaps specified by the `swap_rate` parameter, under control of the swap time window settings.
- `USER_OK`
  Generates lists of swap suggestions approximately once an hour, and then waits for the user to approve the swap list before proceeding.

min_perf_period
Specifies the time period for samples (in hours) required initially before a recommendation is made. Ensure that the values specified are long enough (usually a week) for Optimizer to establish a good characterization of the typical workloads. Keep in mind that the Optimizer statistics database holds about 14 days worth of data.

workload_period
Specifies how far back in time (in hours) Optimizer considers when the optimization algorithm is run. Ensure that this value is not too large, to avoid including old data that is not representative of current workloads.

max_simult_swaps
Specifies how many swaps Optimizer can perform simultaneously (up to four). The value should reflect the number of devices to be swapped simultaneously. Acceptable values are from two (a single pair swapped) to eight (four pairs swapped).

swap_rate
Specfies the maximum number of device swaps that Optimizer is allowed to make in a single day. This parameter is relevant only if `swap_mode` is set to `AUTO`. For example, a value of 24 allows 12 pairs of devices to be swapped in a day.

Example

To analyze data from the previous seven days and to start determining swap suggestions after three days of collecting data, in the command file enter:

Note

User approved mode is set, and the maximum number of simultaneous swaps is set to 8 (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;
```

Set advanced parameters

Syntax

Use the following syntax to specify advanced parameters:

```
set advanced_parms [max_rollback = max_days,]
                   [hot_spot = TRUE | FALSE,];
```

Options

max_rollback
Specifies the maximum number of days for rollbacks. For example, if 8 was specified, only swaps that took place within the past eight days would be eligible for rollback.

hot_spot
Specifies whether the disk access time is computed from the statistics collected for each time sample.

Optimizer time windows

The `symtw` command manages time windows. 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 Managing Time Windows (Enginuity 5876 or lower) on page 605 for information about managing time windows with the `symtw` command.

NOTICE

Use the `symtw convert` command to convert any existing time windows (previously created with the `symoptmz` command) to the new format.

Set the swap priority for devices

Syntax

To specify device swap priority levels, use the following syntax:

```
set swap_priority NO_SWAP | NORMAL | HIGH for
    dev SymDevStart[:SymDevEnd];
```

Options

NO_SWAP

Sets the device to never be swapped.

NORMAL
Allows swaps that improve performance.

HIGH
Ensures the best performance. Use this setting for devices that store critical application data which require the highest possible 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;
```

Set manual swap lists for Optimizer (Enginuity 5874 and 5876)

Syntax

To set a manual swap list, use the following syntax:

```
set dev_swap SymDevName1 with SymDevName2  [, SymDevName3 with
SymDevName4, ...]
  [begin_at=TimeVal];
```

Note

You can set manual swaps between different device protection types.

Example

```
set dev_swap 0030 with 0040;
```

Set manual swap lists for Optimizer (Enginuity 5773 and lower)

Syntax

To set a manual swap list, use the following syntax:

```
set swap_list {Hyper1} with {Hyper2} [, {Hyper3} with {Hyper4},... ]
[begin_at=TimeVal];
```

Options

Hyper*n*
A hyper identifier, in the form `DDD:I:T:HH`. Where `DDD` is the director identifier, `I` is the director interface, `T` is the target ID, and `HH` is the hyper number.

TimeVal
A time, in the form `MMDDYYYY:HHMMSS`.

Reviewing Optimizer plans

This section provides examples of Optimizer-generated plans for swaps.

Show Optimizer parameters for planned activities

Syntax

To show Optimizer's planned activities, use the following syntax:

```
symoptmz show [-v] -swap_list | -activity_list
          [-manual | -generated ]
```

```
       show -parms [-dp | -vp] [-offline]
       show -composite [-dp | -vp]
       show [-v]  -swap_list | -rollback_list
```

Options

activity_list
Displays all the swap lists currently known to Optimizer (interchangeable with `swap_list`).

generated
Displays Optimizer-generated and FAST-generated swap plans.

manual
Displays any user-defined swap plans.

parms
Displays information about Optimizer control parameters.

dp | vp
`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.

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 | vp
`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.

Show Optimizer parameters for thin devices

Example

To show the Optimizer parameters set for array 234 for thin devices only, enter:

```
symoptmz -sid 234 show -parms -vp
```

Sample output

```
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
```

Show Optimizer parameters for all devices

Example

To show the Optimizer parameters set for array 234, for all devices, enter:

```
symoptmz -sid 234 show -parms
```

Sample output

```
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
```

Show all Optimizer composite windows

Example

To show all the composite windows for array 234, enter:

```
symoptmz -sid 234 show -composite
```

Sample output

```
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
```

Show Optimizer swap list

Example

To display the swap list for array 234, enter the following command:

```
symoptmz show -swap_list -sid 234
```

Sample output

```
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
. . .
```

Output fields

The following information is included in the output:

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. 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. Possible values are:

Done
InProgress
NotStarted
Failed
The execution of the entire group has failed and will be retried.

Time Started
The time the swap started.
Time Completed
The time the swap completed.

Show Optimizer generated plans

Examples

To display a list of Optimizer-generated plans for array 234, enter:

```
symoptmz show -swap_list -sid 234 -generated
```

Sample output

```
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 also display in the Optimizer output. There are no plans for thin tiers.

Approve a swap list

Description

Use Optimizer user approval mode to preview which swaps will occur before they take place. To set user approval mode, use the `set control_parms swap_mode=USER_OK;` line in the command file. After this is set, Optimizer recalculates a swap list approximately once every hour as long as samples are collected for the specified minimum performance period.

If the command to approve or decline the swap list returns an error, the swap list might be out of date. Retrieve the latest swap list with the `symoptmz show -swap_list` command. Show Optimizer swap list on page 598 provides example output from the `symoptmz show -swap_list` command.

Syntax

To approve or decline a swap list, use the following syntax:

```
set swap APPROVE | DECLINE
  [begin_at=TimeVal,]
  TIMESTAMP=TimeVal
```

Options

TIMESTAMP

The time stamp returned by the last `symoptmz show -swap_list` command.

Perform Optimizer rollbacks

Description

Rollbacks are used to undo swaps that conflict with the business rules. Optimizer reverses all swaps by going backward from the present to a selected earlier time and undoing each and every swap.

To undo specific swaps, use the User Approved mode as described in Approve a swap list on page 602. Rollback swaps can be scheduled for execution according to normal Optimizer policy.

Use the `symoptmz show -rollback_list` command to view the rollback list.

Note

When Optimizer rolls back a swap, FAST swaps are affected if they are in the swap list. For more information about FAST, see Fully Automated Storage Tiering (Enginuity 5874 through 5876) on page 180.

Syntax

To roll back a swap, Use the following syntax:

```
set swap <APPROVE | DECLINE>
  [begin_at=TimeVal,]
  TIMESTAMP=TimeVal,
  [,ROLLBACK];
```

Read Optimizer logs

Description

Use the `symoptmz read` command to read the Optimizer activity log and determine why a swap was not performed. Note the following:

- To remove DOS line-feed characters from returned logs, use `sed` or another text editor, as follows:

```
cat opt_log.txt | sed s/^M$// > cleaned_opt_log.txt
```

- In a UNIX shell, type the ^M character using **CTRL-V CTRL-M**.

Syntax

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 following command:

```
symoptmz read -log_type ERROR [-start DateTime] [-stop DateTime
```

Optimizer device migration (Enginuity 5773 or lower)

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 Virtual LUN Technology (Enginuity 5874 through 5876) on page 469

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 migrate** command provides the following data migration features:

- The source devices to be migrated can be specified as a device group.
- The target disks can be a disk group number.
- 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, 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. Optimizer does not look for free space on the target disks but looks only 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.

Migrate devices using Optimizer

Syntax

To initiate a migration, use the following syntax:

```
symoptmz migrate
   dev[s] SymDevStart1[:SymDevEnd1]
   [,SymDevStart2[:SymDevEnd2],...]
   TO disk[s] {disk
n
} [,{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];
```

Options

disk*n*
: A disk identifier, in the form `DDD, I, T`. Where `DDD` is the director identifier, `I` is the director interface, `T` is the target ID.

TimeVal
: A time in the form of `MMDDYYYY:HHMMSS`.

Example

```
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 19

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.

- Time windows overview 606
- Time windows conversion 607
- Add time windows 609
- Remove time windows 610
- Display time windows 612

Time windows overview

Note

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 the`symoptmz set time_window`command.).

Both the `symoptmz` and `symtw` commands are valid, however only use of one or the other is allowed. To convert to the`symtw` format use the `symtw convert` command.

Note

After using the `symtw` command, the `symoptmz` command for setting time windows is no longer supported.

Time window types

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.

Time window definitions

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 are 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 has the highest priority and it overrides 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 do not replace the current time windows. They are added on top of the current time windows. The remove operation allows the user to remove any specified time windows.
- Any expired exclusive time windows are 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 are 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 collects performance statistics and execute data movement only after a `symtw convert` command is issued.
- 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 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 be set to a 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 use a Sunday to Saturday week. Therefore, if the command, `symtw convert` is entered 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.

Time windows conversion

The `symtw convert` command converts any legacy time window definitions defined through the `symoptmz` command to the enhanced `symtw` time window format.

If the conversion start 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 is the current day if the exact date information is not provided.

Time Windows Conversion guidelines

Follow these guidelines when converting legacy time windows:

- Execute the `symtw convert` command to convert to the `symtw` command format. All `symtw` commands 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 `symtw convert` command is also required and it sets the database to the enhanced time window format.
- Only the time windows defined during the 7 days including the start date are converted. All inclusive time windows defined before or after this period of time are deleted.
- Exclusive time windows previously defined for a future date convert to the enhanced time window format only if it is defined as `ONCE`.
- The start and end time of the time windows are rounded to the nearest half hour time. The start time is rounded down to the previous half hour and the end time is rounded up to the next half hour.

Time Windows conversion examples

Description

The `symtw convert` command converts 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 are converted to the enhanced time window format.

For example, the `symoptmz -sid 432 -f legacy_windows commit` command was previously run, for the legacy time windows that are in the file `legacy_windows`.

Contents of the file `legacy_windows`, are:

```
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;
```

Examples

To convert the time windows for array `432`, enter:

Note

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.
```

Add time windows

Syntax

To create a time window, use the following syntax:

```
symtw -sid SymmID -inclusive [-noprompt]
          -type <move_dp | move_vp | perf | all>
    add   -days DayList -start_time Time -end_time Timesymtw -sid
SymmID -exclusive [-noprompt]
          -type <move_dp | move_vp | perf | all>
    add   -start_day
DateTime -end_day
DateTime
```

Options

-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 for confirmation.

-type
Indicates the type of time window, as follows:
Possible values are:

- move_dp—Moves data on disk provisioned devices.
- move_vp—Moves data on virtually provisioned devices.
- perf—Gathers performance data.
- all —Creates all three types of time windows.

-days
Specifies the days for the time window. Possible values are: Any comma-separated combination of **MON**, **TUE**, **WED**, **THU**, **FRI**, **SAT**, and **SUN**.

-start 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.

-end 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.

-start day
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**.

-end day
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**.

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
```

Two exclusive time windows 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
```

Remove time windows

Syntax
To remove time windows, use the following syntax:

Note

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.

```
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
```

Options

-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 for confirmation.

-type
Indicates the type of time window, as follows:
Possible values are:

- move_dp—Moves data on disk provisioned devices.
- move_vp—Moves data on virtually provisioned devices.
- perf—Gathers performance data.
- all —Creates all three types of time windows.

-days
Specifies the days for the time window. Possible values are: Any comma-separated combination of **MON**, **TUE**, **WED**, **THU**, **FRI**, **SAT**, and **SUN**.

-start 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.

-end 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.

-start day
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**.

-end day
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**.

Examples

To remove the time windows from 17:30 to 18:30 on Monday, Tuesday, Wednesday, Thursday and Friday, enter:

Note

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 are Monday to Friday and the starting and ending times are from 18:30 to 24:00 for each day.

To remove the exclusive time window, enter:

```
symtw -sid 397 remove -type MOVE_DP -exclusive
    -start_day 11042011:0000 -end_day 11042011:0800
```

Remove all time windows

Description

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.

Examples

To remove all time window definitions, enter:

```
symtw -sid 397 rmall -type ALL -exclusive -inclusive
```

Display time windows

Syntax

To list a defined time window, use the following syntax:

```
symtw [-sid SymmID] [-offline]
   list [-type <move_dp | move_vp | perf>]
   list -summary [-date Date
```

Options

-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**.

Examples

To show the time windows defined for array 397, enter:

```
symtw -sid 397 list
```

Sample output

```
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)
```

List time window definition summary

Description

The `symtw list -summary` command lists 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.

Examples

To show a summary of the time windows definitions for the current week, enter:

```
symtw list -sid 397 -summary
```

Sample output

Time windows definitions for the current week:

```
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 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.

- Network IPsec Overview 616
- IPsec and EMC arrays 616
- SYMCLI IPsec management component 617
- View policy information 618
- Command file scripting for creating/modifying policies 621
- Commit a policy-scripted command file 627
- Delete a policy-scripted command file 628

Network IPsec Overview

NOTICE

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 at the same time. 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 through 5876, 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 selective, 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

It is recommended that IPsec management be performed with or by network engineers that understand encryption disciplines and the standard RFC 2401 for IPsec, as well as specific network and environment details. 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 performs 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.

The Access Control feature with **`symacl`**, requires access types BASE and CFGSYM.

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 displays and sets 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.

IPsec key management

Automatic 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

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.

Internet Key Exchange (IKE) protocol

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.

View policy information

The `symipsec` command, returns a list of IPsec policies or shows 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.

List IPsec policies

Syntax

To list information about policies on a specified VMAX array and director, use the following syntax:

```
symipsec -sid SymmID list -dir #|ALL -port 0 -priority Level# -all
```

Options

-priority

The policy level index number. The lower the number, the higher the priority level (0-110).

Examples

To list the policy priority numbers on director 16D within a specific VMAX array, enter:

```
symipsec -sid 343 -dir 16D list port 0 -all
```

Sample output

This output lists 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
```

Show IPsec policy detail

Syntax

To show detail information about policies on a specified VMAX array, use the following syntax:

```
symipsec -sid SymmID show -dir #|ALL -port #|ALL -priority Level# -all
```

Options

-priority

The policy level index number. The lower the number, the higher the priority level (0-110).

Examples

To show the policy details on director 1A for a specific policy (20), enter:

```
symipsec -sid 123456789012 -dir 1A -port 0 show -priority 20
```

To show the policy details on director 16D for all current policies, enter:

```
symipsec -sid 343 -dir 16D show -all
```

Sample output

This output shows detail information for all current policies on the specified director.

```
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:
.
```

```
.
.
```

List IPsec policy statistics

Syntax

To list statistical information about the IPsec processing of a policy or policies, use the following syntax.

```
symipsec -sid SymmID list -dir #|ALL -port 0 -stats -type StatsType
[-priority Level#][-remote_addr IPendPt
```

Options

-stats -type
Possible values are:

- ike_errors
- ipsec_details

-remote_addr
The IP endpoint address.

-priority
The policy level index number. The lower the number, the higher the priority level (0-110).

Examples

To display IPsec SA statistical details for a specified director 2C, enter:

```
symipsec -sid 067 -dir 2c list -stats -type ipsec_details
```

Sample output

This output lists 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
```

List IPsec IKE errors

Example

To display IPsec IKE errors in the security processor on director 2C, enter:

`symipsec -sid 067 -dir 2c list -stats -type ike_errors`

```
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 is done through scripting and managing a command file. Adding modifying, or deleting policies is performed using various entries in a specified command file.

Command file syntax adding/modifying policies

Syntax

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

Only one proposal and one transform are allowed 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.
;
```

Script IPsec policy syntax

Syntax

The beginning section of the policy syntax sets the overall policy requirements. For a policy add or modify task, use the following syntax:

```
-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
```

Options

-priority
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
Sets the policy action. Possible values are:

- secure - Requests that IPsec processing will be fully applied to packets that match this policy, requiring packet transformation.
- discard - requests that the matching packets will be dropped completely. For use on unauthorized break-ins.
- bypass - requests that the matching packets will be completely unmodified, and pass straight through the IPsec processor without encryption.

-assoc_ike_policy
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.

-remote_tunnel_addr
Specifies the IP address of the remote tunnel object. The -esp_mode option must be set to tunnel.

-selectivity {destip|destport|srcport} SPECIFICITY
Specifies 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.

Define local and remote endpoint IP addresses

Syntax
To define local and remote endpoint IP addresses, use the following syntax:

Note

The SYMCLI can detect and handle both IPv4 and IPv6 address notations (dots and/or colons) in the command entry.

```
-local_addr IPaddr[-ipproto IPprotocol#|all][-ipport IPport#|all][-
mask IPaddr]
-remote_addr IPaddr[-ipproto IPprotocol#|all][-ipport IPport#|all][-
mask IPaddr]
```

Options

IPprotocol#
Targets an IP protocol number. For example, IP protocol 6 should be used for TCP and IP protocol 1 for IPv4-ICMP.

IPport#
Targets an IP port number. For example, IP port 1748 should be used for SRDF and IP port 3260 for iSCSI host connections.

Script proposal set syntax

Syntax

The proposal set section of the policy syntax sets the proposal settings requirements. To add or modify a policy, use the following syntax:

```
-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
```

IPsec manual options

Syntax

```
-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
```

Options

These options require **proposal_set_type** to be set to manual.

-enc_alg
Use for non-null.

-inenc_key
Specifies an encryption key string used for encrypting/decrypting inbound traffic. This must match the corresponding field on the remote endpoint.

outenc_key
Specifies an encryption key string used for encrypting/decrypting outbound traffic. This must match the corresponding field on the remote endpoint.

-auth_alg
Use for non-null.

-inauth_key
Specifies a hash key string used for authenticating inbound traffic. This must match the corresponding field on the remote endpoint.

-outauth_key

Specifies a hash key string used for authenticating outbound traffic. This must match the corresponding field on the remote endpoint.

-in_spi
Specifies a Security Parameter Indexing (SPI) number for inbound traffic in security associations decode.

-out_spi
Specifies a Security Parameter Indexing (SPI) number for outbound traffic in security associations decode.

-in_nonce
Specifies a random nonce value for inbound traffic to counter replay attacks.

-out_nonce
Specifies a random nonce value for outbound traffic to counter replay attacks.

IPsec IKE Options

Syntax

```
-ike_mode main|aggressive
-pfs on|off
-presharedkey
Keystring
```

Options

These options require **proposal_set_type** to be set to **ike**.

-ike mode
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 - more intense and secure but time consuming
- aggressive - provides faster negotiations, but exposes identities of the peers to eavesdropping.

-pfs
Perfect Forward Secrecy (PFS) mode for IKE policies can be turned on or off. Typically this should be left on except for special environments.

-presharedkey
Specifiies the preshared key as the authentication method and defines the preshared key string. It should be the same secret string shared between security points

Proposal options

Syntax

These options start the proposal declaration. Currently, only one proposal per policy can be declared. Compression mode is not supported.

```
-proposal
-proposal_type ike|ipsec
```

Transform options

Note

DSA and RSA methods are not supported.

Start the proposal declaration

```
-transform
-transform_type ike|esp
```

These options start the transform declaration. Currently, only one transform per proposal and policy can be declared.

-transform_type
Specifies the type of transform to apply to the policy proposal. Supported types are:

- ike - IP Key Exchange
- esp - Encapsulation Security Payload

Authentication header (ah) is not supported.

Specify authentication algorithm for IKE or ESP policy and transform hash functions

```
-auth_alg null|sha1|md5|xcbc
```

-auth_alg
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

Specify the encryption algorithm for IKE or ESP

```
-enc_alg null|des|3des|aes_128|aes_256|aes_cm_128|aes_cm_256
```

-enc_alg
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)

Specify Diffe-Hellman (dh) group for symmetrical key generation for IKE transforms

```
-dhgroup 1|2|3|4
```

-dhgroup
Specifies which Diffe-Hellman (dh) group to use for the symmetrical key generation for IKE transforms only. Groups 1 through 4 are supported.

Specify the ESP transform mode

```
-esp_mode tunnel|transport
```

-esp_mode
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.

Specify the life of the security policy

```
-lifetime [
LifeParam1][,][
LifeParam2
```

-lifetime
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, are specified as:

- 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.

Authentication method for IKE transforms

```
-auth_method preshared_key
```

-auth_method
Specifies the authentication method for IKE transforms. Possible values are **preshared_key**

Commit a policy-scripted command file

After scripting the policy in the command file, commit the file for execution. Due to extensive notation work in any policy command file, EMC recommends previewing the file for command syntax errors.

Preview the IPSEC command file

Syntax
To preview the IPsec policy syntax of a command file, use the following syntax:

```
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.

Example
To target director 1A to preview an IPsec policy addition to the policy database, using file `/tmp/ap`, enter:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/ap preview
```

Commit IPSEC command file

Syntax

To commit the IPsec policy syntax of a command file, use the following syntax:

```
symipsec -sid SymmID -dir # -port #  -file FileName commit
```

Examples

To target director `1A` to commit an IPsec iSCSI policy addition to the policy database, using file `/tmp/ap`, enter:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/ap commit
```

Where the file **`/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;
```

Delete a policy-scripted command file

Examples

To remove an IPsec policy from the policy database specified in a file, enter:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/dp delete
```

To remove an IPsec policy from the policy database using the policy priority level number, enter:

```
policy delete -priority 50;
```

CHAPTER 21

Events and Logs

This chapter describes how to configure, manage, and query VMAX events and logs.

- Events and logs overview 630
- Log option configuration 630
- VMAX event monitoring using SYMCLI 632
- Common audit log 633
- Event daemon 635

Events and logs 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
```

File name date format:

- *yyyy* — year
- *mm* — month
- *dd* — day

The log displays the following items about 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*.

Log option configuration

Table 56 on page 630 lists the options to customize how and where events are logged in your VMAX environment.

Table 56 Log file configuration options

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`

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:

Table 56 Log file configuration options (continued)

Logging option	Configuration description
	`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 the 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.

VMAX event monitoring 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 monitoring of the VMAX array for all reported events.

Use **`symevent`** to:

- `monitor` — This action runs the command 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. Monitoring can be restricted to report events of certain severity (warnings, errors, or fatal events).
- `list` — This action examines the history of events stored on the VMAX array, for those events that meet the requested criteria, such as events recorded during a certain time period, or based upon the reporting director.

Monitoring events

Examples

To poll for and report on all events, on all locally-connected VMAX systems, every 15 seconds, continuously, enter:

```
symevent monitor
```

To poll for and display events with a severity of *warning* or greater every 15 seconds for a 50-second period, enter:

```
symevent monitor -sid 0207 -i 15 -c 50 -warn
```

Sample output

For array `0207`:

```
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
```

List events

Options

-start / -end
Lists events occurring during a certain time period.

-DIR
Limits reporting to specific director activity.

Examples

To retrieve a verbose list of the events which have occurred on the specified array between 12 p.m. and 3 p.m. today, enter:

```
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. 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 filtering of the common audit log file for a specified VMAX array. The audit log resides on the array with a capacity of 40 MB. Once the 40 MB limit is reached, the log starts to overwrite itself. There is no maintenance for this file, unless records need to be captured before the circular 40 MB space recycles.

Note

For details on filtering parameter values refer to the `symaudit` command in the *EMC Solutions Enabler SYMCLI Command Reference Guide*

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
- device acted upon

monitor
Monitors the array for new audit log data in realtime.

Note

For HYPERMAX OS 5977, if Embedded Management is configured on the VMAX array, when running the `symaudit list` and `symaudit show` commands the host name listed in the output is the nodename identified by the NAT gateway, and not the internal identity of the eManagement Guest. For more information on eManagement refer to the *EMC VMAX3 Family Product Guide*.

Show audit log for single array

Examples

```
symaudit show -sid 0207
```

Sample output

```
    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
```

List audit log for specified time period

Examples

```
symaudit list -sid 0207 -start_date 01/02/2009:12:00:00 -end_date
01/02/2009:12:15:00 -v
```

Sample output

```
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; ReserveID=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
<. . .>
```

Output field descriptions:

- `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 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).
  For more information on 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.

- XML structured output overview 638
- XSLT: XML data transformations 638
- Element-based XML 638
- Set XML mode with SYMCLI 639

XML structured output 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.

Set XML mode with SYMCLI

To use XML mode with SYMCLI, an environment variable or a command line option can be used. To use the environment variable to globally set the command output to XML or standard use the following syntax:

```
SYMCLI_OUTPUT_MODE = xml_element|standard
```

Note

When the environment variable output mode is set to **xml or xml_element**, commands that do not support XML output generate a runtime error message. To override this behavior set the command line **-output** option to a value of **standard**. This allows execution of a given command in standard mode.

XML output using SYMCLI

Syntax

To override the current environment variable setting and set the output style for a single command, use the following syntax:

```
<SymcliCommand> -output <xml|xml_element|standard>
```

Note

The `-output` flag is not found in `-help` or man pages because of its wide scope and usage.

Options

xml_element
Returns the output of all commands in element-based XML tags.

standard
Returns the output of all commands in the default output format.

Examples

```
symcfg list -out xml or symcfg list -out xml_element
```

Sample output

Note that a new element tag **<Symm_Info>** is added to the XML data shown below to store general VMAX data.

```
<?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

This section describes how to use:

- Open Replicator (symrcopy) to migrate data between EMC arrays or third-party arrays
- Federated Live Migration (symrcopy -migrate) to migrate data from DMX arrays to VMAX arrays.

Chapters include:

Chapter 23, "Open Replicator Operations"

Chapter 24, "Operational Restrictions and State Reference"

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

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 644
- Open Replicator command summary 651
- Open Replicator session options 652
- Open Replicator control operations 655
- Open Replicator examples 680

Open Replicator overview

The Open Replicator `symrcopy` command 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 used to migrate data from older arrays, and some third-party storage arrays to a VMAX array. The **`symrcopy`** command is part of the SYMCLI command library.

The Open Replicator command is also 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 comman

NOTICE

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

ORS 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.
- Listing and terminating RecoverPoint sessions.
- 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 Federated Live Migration Operations (Enginuity 5876 or lower) on page 725 for more information.

  Note

  For arrays running HYPERMAX OS 5977 or higher, FLM is not supported.

ORS 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.

ORS and host interaction

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.

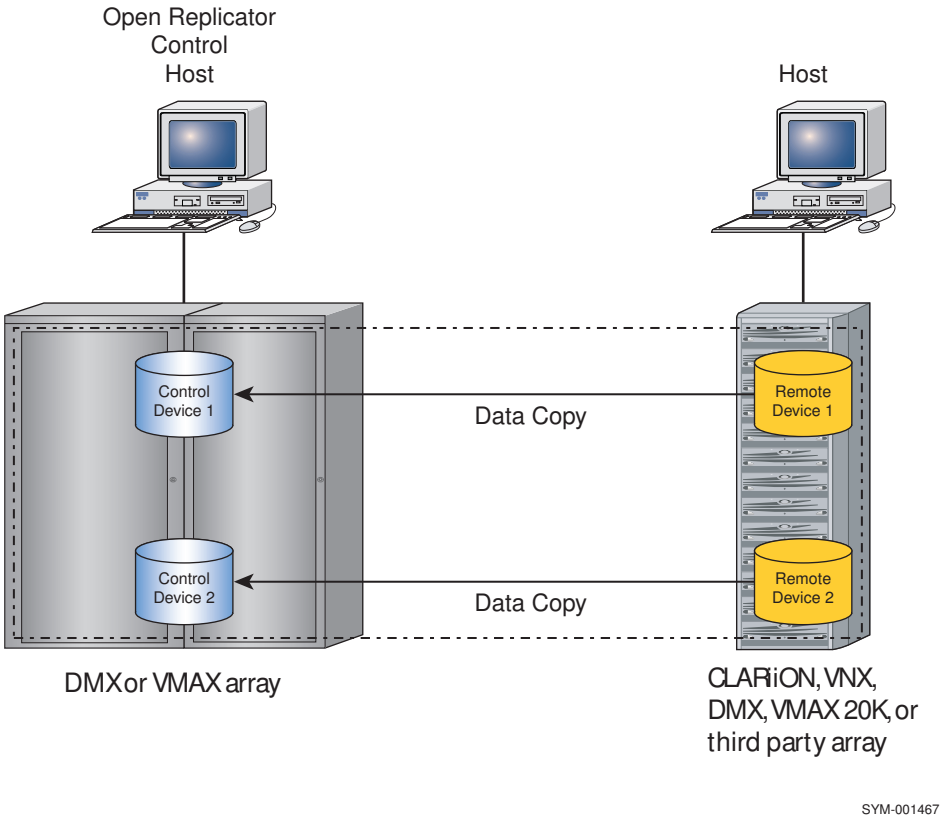
ORS rcopy concepts

The following Rcopy concepts and terminology are used throughout the Open Replicator Migration section of this product guide:

- The VMAX 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 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 21 on page 646 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 21 Control array device pull operation



Note

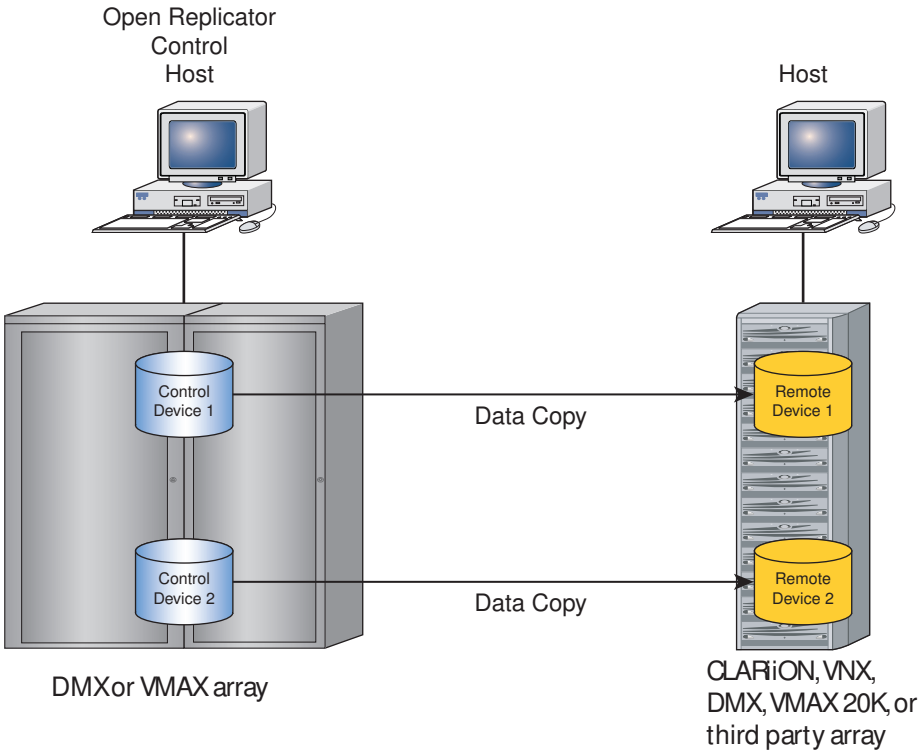
Since data is copied through the SAN infrastructure, Open Replicator may require updating the zoning configuration before copying data between arrays is allowed. For zoning requirements and suggestions, refer to ORS SAN setup requirements on page 650.

Open Replicator can be used to copy data from a control array to older array. Figure 22 on page 647 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.

Note

For arrays running HYPERMAX OS 5977, push operations are not supported.

Figure 22 Control array device push operation



ORS 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 is 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. 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 331.
- Open Replicator supports copy operations for a cold push from a virtual device (VDEV). For more information and instructions, refer to VDEVs in Open Replicator on page 678.

ORS 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.

ORS copying limitations

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.

ORS Federated Tiered Storage support

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 361.

Restrictions

- 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.

ORS device guidelines

Table 57 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=60060480000000003143535 94D303737 Refer to ORS and creating a device file on page 656 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	One device per session Device Not Ready to host	Can use **`-nodifferential`**

Table 57 Control and remote device guidelines (continued)

Action	Control device	Remote device
HYPERMAX OS 5977)	At least one director must see remote device(s)	**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.

ORS 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.
- 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.
- For arrays running HYPERMAX OS 5977:
  - Dynamically finds directors and ports that can access an ORS session's remote devices, and uses these directors and ports for the data transfer.
  - 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.

ORS SYMCLI symsan support

The SYMCLI command **symsan** lists port and LUN WWNs as seen from a specific array director and port. This is 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 are seen by a specific DX director or all DX directors.
- Listing all ports or LUNS in the SAN that are seen by a specific FA director or all FA directors.
- Listing all ports or LUNS in the SAN that are seen by a specific FA/DX director or all FA/DX directors, using the `-dir` option.

Refer to SYMCLI help options on page 30 for information on how to access `symsan` manpage or command help.

Open Replicator command summary

Command — symrcopy arguments

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
Creates an incremental copy session on an existing copy session. Only valid for push copy sessions created with differential copying. Not supported for arrays running HYPERMAX OS 5977 or higher.

remove
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
Restores data from a remote device that was previously copied during a differential push operation back to the control device. Not supported for arrays running HYPERMAX OS 5977 or higher.

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
Sets the session pace for the CopyInProg, RecreateInProg, and RestInProg states. Possible value are 0-9, with 9 being the slowest pace. Not supported for arrays running HYPERMAX OS 5977 or higher.

terminate
Terminates a copy session and removes it from the control array.

verify
Verifies that a copy session is in an existing state.

Command — symqos -rcopy arguments

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.

Open Replicator session options

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 ORS and creating a device file on page 656 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`) define the direction of the copy operation for device pairs listed in the device file, and the hot/cold options (`-hot|-cold`) define online or offline copying.

symropy session options

-pull
Pulls data through the SAN to the control device(s) from the remote device(s).

-push
Pushes data across the SAN from the control device(s) to the remote device(s)

-hot
Control device is available as 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.

-cold
Control device is unavailable 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.

Note

Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and uses these directors and ports for data transfer.

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.

ORS hot pull copy session example

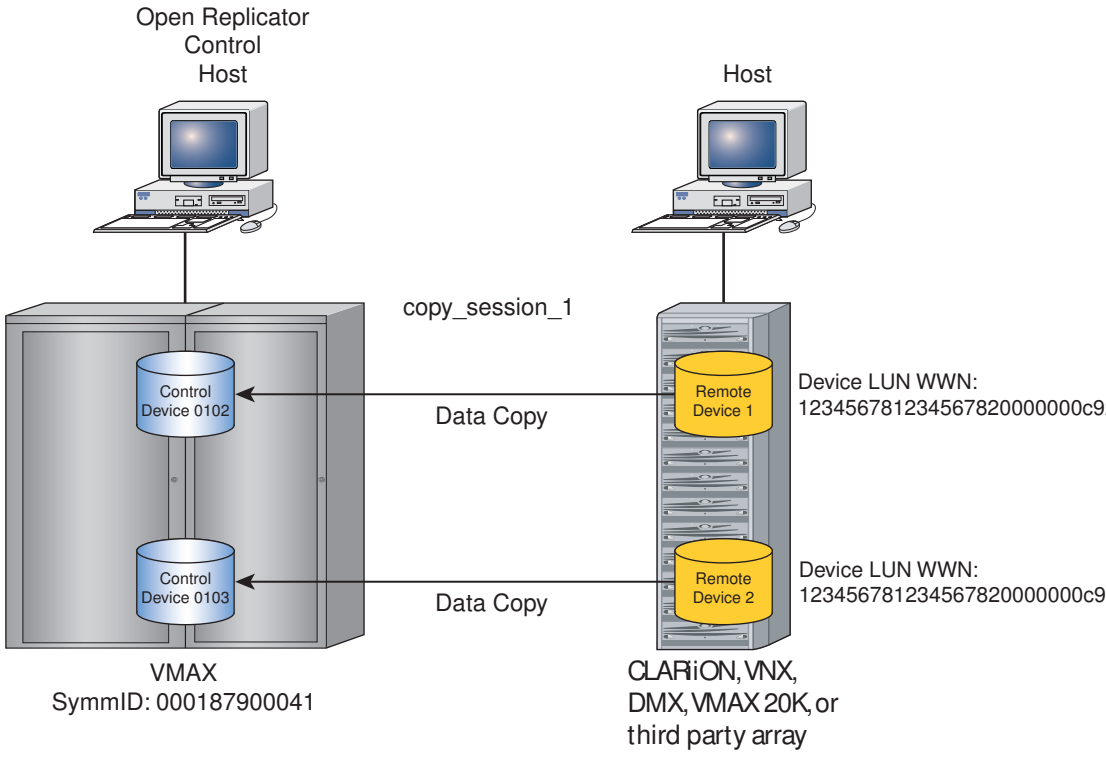
Figure 23 on page 654 shows Open Replicator copy sessions created and activated for a hot pull copy operation. A device file `–file` *`filename`* contains the pairing information for the control and remote devices. Each line in the device file is a copy session. The file specifies control devices by "Symmetrix ID: device number" and remote devices by "LUN WWN" as follows:

```
symdev=000187900041:0102    wwn=12345678123456782000000c920b484
symdev=000187900041:0103    wwn=12345678123456782000000c9274156
```

Note

Refer to ORS and creating a device file on page 656 for instructions on how to obtain device information and create the device file.

Figure 23 Control array hot pull using the symrcopy command




```
symrcopy create -name copy_session_1 -pull -hot -file pairs
symrcopy activate -file pairs
```

ORS cold push copy session example

Figure 24 on page 655 shows Open Replicator copy sessions created and activated for a cold push operation. A device file (`-file filename`) contains the pairing information for the control and remote devices. Each line in the device file is a copy session. The file specifies control devices by "Symmetrix ID: device number" and remote devices by "LUN WWN" as follows:

```
symdev=000187900041:0102    wwn=12345678123456782000000c920b484
symdev=000187900041:0103    wwn=12345678123456782000000c9274156
```

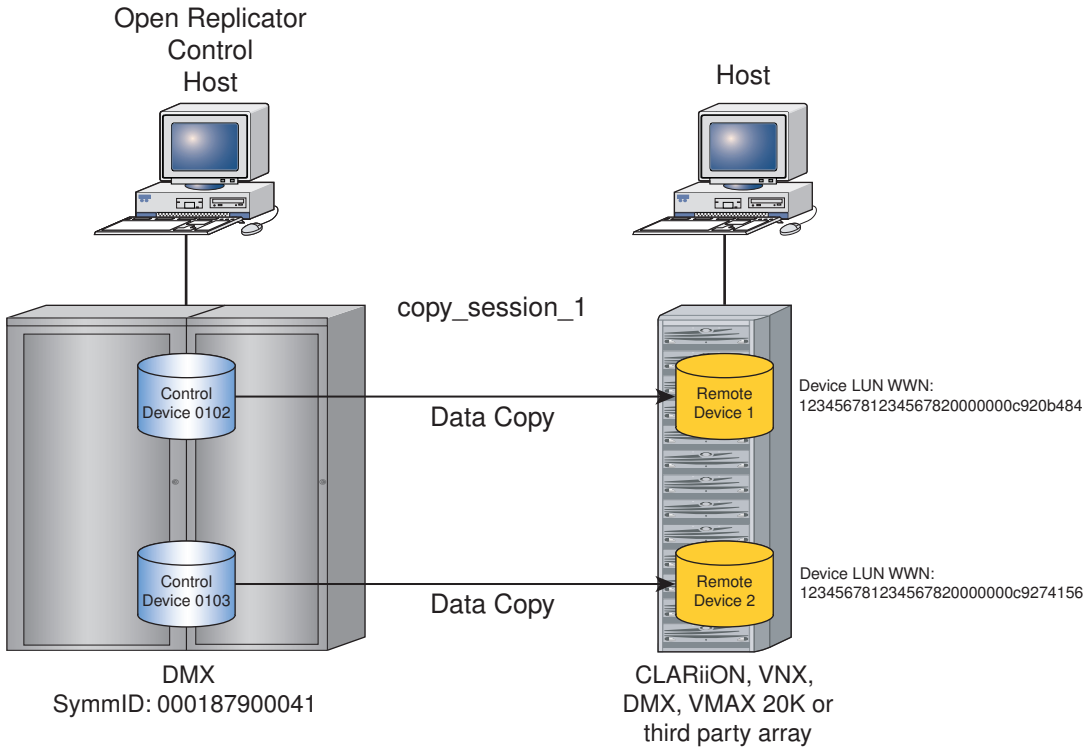
Note

Refer to Obtain device information on page 656 and Create ORS device file on page 658 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.

Figure 24 Control array device cold push using the symrcopy command




```
symrcopy create -name copy_session_1 -push -cold -file pairs
symrcopy activate -file pairs
```

SYM-001470

Open Replicator control operations

SYMCLI Open Replicator performs control operations from a local host attached to the DMX or VMAX Family array and implements these operations 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.

Note

For arrays running HYPERMAX OS 5977, only one remote target is supported.

Open Replicator SYMCLI **symrcopy** command performs the 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.

HYPERMAX OS 5977 settings and actions

- 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.

Enginuity 5876 or lower settings and actions

- 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*

ORS and 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 (one for control devices and one for remote devices). Devices in the file are specified either by their unique LUN WWN, or by the storage array ID and device number (Storage ID: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.

Obtain device information

Description

The Solutions Enabler (SYMCLI) uses several commands to obtain device information, including device number, director information, WWN, and capacity. This information is

helpful in determining and identifying devices for use in a device file. Some of these commands include: `symdev`, `syminq`, `sympd`, `symsan` and `symstat`.

Examples
To list of devices on array `041`, enter:

```
symdev list -sid 0041
```

To list devices by device wwn (a SCSI inquiry), enter:

```
syminq -sym -wwn
```

Sample output
For array `041` devices:

```
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
<. . .>
```

For device wwn list:

```
syminq -sym -wwn
Device                                          Device
------------------------------ ---------------- --------------------------------
Name                         Num    Array ID         WWN
------------------------------ ---------------- --------------------------------
/dev/vx/rdmp/c5t0d2s2      00168 000000006190      6006048000019030001653303030323 8
/dev/vx/rdmp/c5t0d3s2      001F8 000000006190      6006048000019030001653303030323 9
/dev/vx/rdmp/c5t0d4s2      001F9 000000006190      6006048000019030001653303030324 1
/dev/vx/rdmp/c5t0d5s2      00170 000000006190      6006048000019030001653303030324 2
/dev/vx/rdmp/c5t0d6s2      00172 000000006190      6006048000019030001653303030324 3
/dev/vx/rdmp/c5t0d7s2      001B2 000000006190      6006048000019030001653303030324 4
```

Note

For details about the `symdev` and `syminq` commands syntax, refer to the *EMC Solutions Enabler CLI Command Reference*

Create ORS device 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.

This example shows a device file with multiple 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
```

This example shows a device file 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
```

This example shows a device file with a mix of symdev and wwn device IDs.

```
symdev=000000001234:01 symdev=000000005678:42
symdev=000000001234:02 symdev=000000005678:43
symdev=000000001234:03 wwn=60060480000000000567853594D303434
```

Export ORS device list to text file

Examples

To export session device list to text file `dev_file_1.txt`, enter:

```
symrcopy export -session_name rcopy_1 -file dev_file_1.txt
```

Sample output

The output file (`dev_file_1.txt`) contains the session device list.

```
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
```

Create ORS copy session

Description

The `symrcopy create` defines a new ORS copy session. command Other mandatory syntax session controls 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*). See Open Replicator session options on page 652 for more information on the mandatory parameters.

Syntax

```
symrcopy create -name <SessionName> <-push | -pull> <-hot | -cold> -
file <FileName>
```

Options

-name
Session name — used for control operations.

Examples

To define a hot, pull Open Replicator copy session named `rcopy_1`, enter:

```
symrcopy create -name rcopy_1 -pull -hot -file dev_file_1
```

Note

Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and uses these directors and ports for data transfer.

ORS precopying

Description

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.

Note

The `-precopy` option does not need to be specified in the `symrcopy create` command. Refer to ORS background copying on page 665 for more information on background copying mode options.

Syntax

To define an Open Replicator copy session using the **`-precopy option, enter:`**

```
symrcopy create -name rcopy_1 -precopy -file dev_file_1
```

NOTICE

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 ORS donor update on page 663.

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. Use the `recreate` command to incrementally update from the control to remote devices. Refer to Recreate an ORS session on page 675 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 an ORS session on page 677 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.

ORS hot copying

Open Replicator copy sessions running online (-hot) are 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.

Restrictions and operating behavior

The following restrictions and operating behavior apply to hot copying:

- For Enginuity 5876 or lower, there is a limit of 32 directors mapped to a device involved in an ORS hot session. If a device is mapped to more then 32 directors, the session creation fails.

  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 Recover from failed ORS session on page 676 for more information.

- 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.
- For arrays running HYPERMAX OS 5977, only one remote target is supported.
- A hot copy session that is copying data from the host control array to other devices located in the SAN (-push), is 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 must 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.
- 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.
- If there is a network failure while hot copying from a control array to other devices in the SAN (-push), the copy session fails. However, the local host can continue writing to the control device without impact to the application.

NOTICE

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 ORS donor update on page 663.

ORS cold copying

With an offline (-cold) copy session, the control device is not available to the host while the copy operation is in progress.

Restrictions and operating behavior

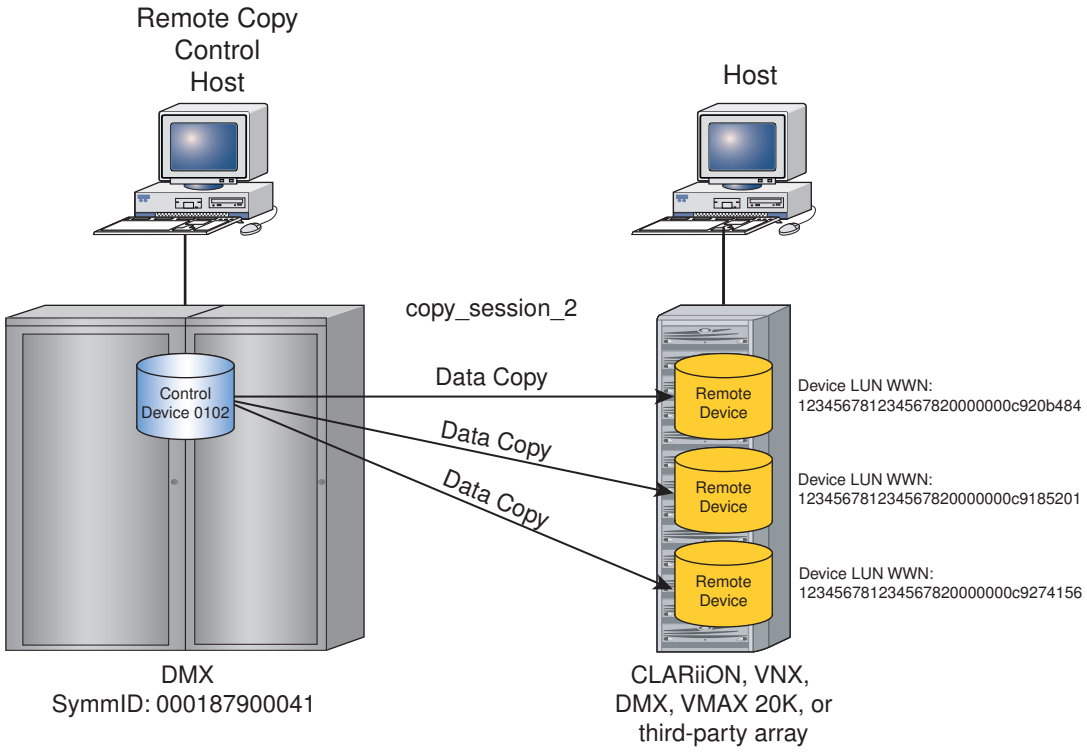
The following restrictions and operating behavior apply to cold copying:

- Session uses directors that are mapped to the control device and can reach the remote device.
- Session is created as long as one or more directors have discovered the remote device during the discovery.
- Control device must be disabled for both reads and writes *(Not Ready)* to the control host.
- Concurrent data copying from one control device to up to 16 remote devices, is only allowed with a cold push operation.
- For arrays running HYPERMAX OS 5977, only one remote target is supported.
- 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 performing a cold pull operation, the session stalls and retries.

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 Recover from failed ORS session on page 676 for more information.

Figure 25 Cold push copy session: single device to multiple remote devices




```
symrcopy create -name copy_session_2 -push -cold -file multi_tgt
symrcopy activate -file multi_tgt
```

SYM-001471

ORS front-end zero detection

Front-end zero detection is an option 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 (thick) control devices.

Note

With front-end zero detection enabled, "persistent" allocations are treated as regular allocations, and track group is de-allocated.

Enable front-end zero detection

Example

By default pull sessions are disabled for the `-frontend_zero` detection option. To create a hot pull session and enable front-end zero detection, enter:

```
symrcopy create -session_name rcopy_1 -pull -hot -frontend_zero -file
dev_file_1
```

Disable front-end zero detection

Examples

The `set frontend_zero off` option disables zero detection for the session, and is only allowed during active **`-pull`** operations. To disable `front-end zero detection` for the active session `rcopy_1`, enter:

```
symrcopy -session_name rcopy_1 set frontend_zero off
```

Note

Once front-end zero detection is disabled during an active session, it cannot be enabled again during the session.

ORS 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.

Enable ORS donor update

Examples

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 Activate ORS session on page 665.

Terminate and restart ORS hot pull session

Examples

If during an activated hot pull operation, a SAN failure or other connectivity issue is detected, then terminate the Open Replicator sessions. To terminate an Open Replicator sessions associated with the control device:

```
symrcopy terminate -file dev_file_1 -symforce
```

To start the copy session again after the problem is resolved, and restart the copy process from the point of failure:

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.

```
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.

Disable ORS donor update

Description

The `set donor_update off` option disables the `donor_update` option. This command stops the copying of data to the remote devices, and stops all new writes to the control device from immediately being copying to remote device. The donor update option may also be used on an incremental restore session. Refer to Restoring an ORS session on page 677 for more information.

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.

Options

-consistent

Maintains consistency of the data on the remote device. Without this option, donor update is still deactivated, but consistency on the remote devices is not maintained. 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.

Examples

To set the donor update option to off and maintain consistency on the remote devices for session `rcopy1`, enter:

```
symrcopy set donor_update off -session_name rcopy_1 -consistent
```

Restrictions:

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.

Activate ORS session

Description

The `symrcopy activate` command activates the copy sessions for device pairs listed in the device file and begins copying data to (pushing) or from (pulling) the remote devices.

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.

Syntax

To begin the copying process for an Open Replicator copy session, use the following syntax:

```
symrcopy activate -file Filename -session_name SessionName
```

Note

Any other Open Replicator copy sessions that were previously created using the specified device file (and session name) will also be started.

Options

-file *Filename*
The device file.
-session_name *SessionName*
The session name.

Example

To activate an Open Replicator copy session, enter:

```
symrcopy activate -session_name rcopy_1
```

Note

Under certain circumstances, failed sessions may be reactivated. Refer to Recover from failed ORS session on page 676.

ORS 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 immediately begincopying 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 Monitor ORS session status on page 668 for more detail.

Note

The `-precopy` option requires Enginuity version 5772 through 5876.

Stop and restart ORS background copying

Examples

To temporarily stop the background copying for a session by changing the state to *CopyOnAccess* or *CopyOnWrite* from *CopyInProg* using the `symrcopy` command, enter:

```
symrcopy set mode nocopy -file dev_file_1
```

To resume background copying for a session and change to the CopyInProg state, enter:

```
symrcopy set mode copy -file dev_file_1
```

Note

The `-precopy` option requires Enginuity version 5772 through 5876.

ORS background copying (hot) without point-in-time copy

Examples

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
```

```
symrcopy create dev_file_1 -precopy
```

Note

To see the `-precopy` option used with the `recreate` command, refer to Recreate an ORS session on page 675.

Restrictions:

- The `-precopy` option requires Enginuity version 5772 through 5876.
- Precopy mode can only be set when the session is not activated.

ORS 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).

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.

Note

The DISABLE setting is only allowed for HYPERMAX OS 5977 or higher.

Set ORS ceiling

Examples

To set a bandwidth ceiling of 100% for all directors on array 6190, enter:

```
symqos -rcopy set ceiling 100 -dir all -sid 6190
```

To view the ceiling setting, enter:

```
symqos -rcopy list ceiling -dir all -sid 6190
```

Set the ORS session pace

Description

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.

Example

The following example shows how to set the session copy pace:

```
symrcopy set pace 0 -file dev_file_1
```

Restrictions:

- The session pace is ineffective to the copy if the ceiling is set to a value other than NONE.
- For arrays running HYPERMAX OS 5977 or higher, setting session pace is not supported.

Monitor ORS session status

Open Replicator session status is checked using the `symrcopy query`, `symrcopy list`, or `symrcopy verify` command.

List all ORS sessions

Description

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.

Options

The following options are available for the `symrcopy list` command:

-offline
: It only displays information held in the database and does not query the array for updated session information.

-detail
: It displays additional device information for modified tracks, session pace, and session name.

-wwn
: It 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.

Example

The following is a list example of all Open Replicator sessions for array ID 6190:

```
symrcopy list -sid 6190
```

Sample output

```
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.
```

List filtered ORS session types

Description

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 List FLM session on page 736.

Examples

To list standard sessions only, enter:

```
symrcopy list -sid 6190 -type standard
```

To list RecoverPoint sessions only, enter:

```
symrcopy list -sid 90 -type recoverpoint
```

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 List all ORS sessions on page 668.

Sample output

For standard session:

```
Symmetrix ID: 0000000006190
          Control Device             Remote Device                   Flags    Status
Done
          ---------------------- ------------------------------- -------  --------
-----
                          Protected
```

```
           SID:symdev   Tracks      Identification                    RI CDSHUTZ SRC <=> TGT
(%)
           ----------- --------- -------------------------------- -- ------- ------------
----
           001F8              33000 6006048000000000619053594D314640 .W X..XXS. CreateInProg
N/A
           001F9              33000 6006048000000000619053594D314637 .W X..XXS. CreateInProg
N/A
           00170              20000 6006048000000000619053594D314642 .W X..XXSX CopyInProg
50
           00172              30000 6006048000000000619053594D314646 .W X..XXSX 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.
```

For RecoverPoint session:

```
Symmetrix ID: 000000006190

           Control Device            Remote Device                   Flags     Status
Done
           --------------- ------------------------------------- --------    ------
-----

Protected
      SID:symdev      Tracks    Identification                    RI CDSHUTZ SRC <=>
TGT    (%)
     ------------   --------- --------------------------------   -- ------ --------------
----
      01B2                33000 6006048000000000619053594D314649  SW ...XXR.
Created        75

          Total        ---------
           Tracks          33000
            MB(s)              2062.5
```

Query ORS session status

Description

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.

Options

The following options are available for the `symrcopy query` command:

-i
: This (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
: This (count) is used with the `-i` option and specifies the duration of the query intervals.

Note

Estimated time to copy completion is shown in the query output when the `-c` and `-i` options are used or if the protected track count has changed since the last interval.

-offline
: This option displays only information held in the database and does not query the array for updated session information.

-detail
: This option displays additional device information for modified tracks, session pace, and session name. This opiton expands the width of the character display, which may not view properly for some displays.

-wwn
: This option displays the full device world wide name. This opiton expands the width of the character display, which may not view properly for some displays.

-summary
: This 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.

Examples

To query copy session status using a file, enter:

```
symrcopy query -file dev_file_1
```

To query for copy session status that will run every 30 seconds for 1 hour, enter:

```
symrcopy query -file dev_file_1 -i 30 -c 120
```

To query copy session status using a session name, enter:

```
symrcopy query -session_name rcopy_2
```

To query copy session status using the `-summary` option, enter:

```
symrcopy -file dev_file_1 query -summary
```

Sample output

This output shows a 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 6006048000000000619053594D314638 .W X..XXM. Copied         100
000000006190:001F8       33000 6006048000000000619053594D314640 .W X..XXS. CreateInProg   N/A
000000006190:001F9       33000 6006048000000000619053594D314637 .W X..XXS. CreateInProg   N/A
000000006190:00170       20000 6006048000000000619053594D314642 .W X..XXSX CopyInProg      50
000000006190:00172       30000 6006048000000000619053594D314646 .W X..XXSX CopyInProg      75
000000006190:001B2       33000 6006048000000000619053594D314649 .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.
```

Verify ORS session state

Syntax

To verify session status, use the following syntax:

```
symrcopy verify [-createinprog | -created | -recreateinprog |
-recreated | -copyinprog | -copyonaccess | -copyonwrite | -copied |
-terminateinprog | -failed | -verifyinprog | -restored | -restinprog
| -precopy
[-cycled] | -syncinprog | -synchronized | -failedback]
```

Options

-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.

-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.

Examples

To verify copy session states for a list of devices specified in `dev_file_1`, enter:

```
symrcopy -file dev_file_1 verify
```

To verify copy session states for a list of devices specified in `dev_file_1`, enter:

```
symrcopy -file dev_file_1 verify -summary
```

Sample output

Using the `verify` command:

```
One of the device(s) in the list are in "~Copied' state.
```

Using the `verify` command with the `-summary` option:

Note

The one-line `verify` command output displays 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.

Terminate an ORS session

Syntax

To terminate an ORS copy session and remove it from the array, use the following syntax:

```
symrcopy terminate
```

Options

-symforce

This option is mandatory when terminating a session in the *CopyInProg*, *CopyOnAccess*, or *CopyOnWrite* state.

NOTICE

Use care when applying the `-symforce` option to terminate an active session. At termination, the receiving devices contain an incomplete data copy and should be considered invalid.

-file

Use this with `FileName` to specify the control device associated with the session. Remote devices in the file are ignored.

-session name

Use this with `SessionName` to specify the session to terminate.

-all sessions
Terminates all sessions.

Examples
To terminate a copy session associated with control device in file `dev_file_1`, enter

```
symrcopy terminate -file dev_file_1
```

To terminate a copy session named `rcopy_1`, enter:

```
symrcopy terminate -session_name rcopy_1
```

To terminate an activated copy session that has not finished copying:

```
symrcopy terminate -file dev_file_1 -symforce
```

To terminate all sessions associated with the control device in file `dev_file_1`, enter:

```
symrcopy terminate -all_sessions -symforce -file dev_file_1
```

Terminate ORS RecoverPoint session

Description
The `symrcopy` command does not support control of Open Replicator RecoverPoint sessions, however for cleanup purposes RecoverPoint sessions can be terminated and removed.

Example
To terminate a RecoverPoint session associated with the control device in the file `dev_file_1`, enter:

```
symrcopy terminate -file dev_file_1 -rp
```

Remove an ORS remote device from a session

Example
When removing a remote device from a session, it must be put in a device file. To remove a remote device associated with device in the file `dev_file_1`, enter:

```
symrcopy remove -file dev_file_1
```

Recreate an ORS session

Description
Recreating a session creates a new point-in-time copy of the data. For differential push operations only, the copy session is 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 changed since the last time the session actively finished copying. Up to 15 sessions are allowed 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.

Options

-name
Use this option to rename a recreated session.

-precopy

Use the option to recreate a copy session to pre-copy the incremental track updates in the background without activating the session. This option is for hot push ORS operations.

Examples

To recreate and activate a copy session for incremental track updates for session name `rcopy_2`, enter:

```
symrcopy recreate -name rcopy_2 -file dev_file_3
symrcopy activate -session_name rcopy_2
```

To recreate and rename a session to `rcopy_2`, enter:

```
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. See Set the ORS session pace on page 667.

Recover from failed ORS session

Description

For a *Failed* session, when no new data is indicated on the devices in the session, it is eligible to be reactivated. Use the `symrcopy query` command to check the status of a failed session. An asterisk (*) symbol next to the `Failed` status indicates the session is available for reactivation. Use the `activate` command to reactivate the failed sessions.

Examples

To query session status, enter:

Note

The *Failed* sessions shown are eligible for reactivation.

```
symrcopy -file dev_file_1 query -detail
```

Sample output

```
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  0060480000000000619053594D314640     SD     X.XX.S.  Failed (*)  N/A
5    N/A
001F9  33000        0  0060480000000000619053594D314637     SD     X.XX.S.  Failed (*)  N/A
5    N/A
Total ---------
 Tracks   66000
 MB(s)        4062.4

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.
```

Restrictions

- Reactivating failed sessions requires Enginuity version 5773 or higher.
- If there is new data on devices that are part of the session, session activation is blocked. If session activation is blocked and it is a *non-differential* session, terminate the session and re-issue the `create` and `activate` commands. This starts 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.

Restoring an ORS session

Description

Restoring a session if only allowed for differential push operations. The copy session restores 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.

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 are set to *Not Ready* status. If running a hot session, control devices are returned to *Ready* status at the end of the operation (as the data begins copying). If running a cold session, the control devices remain in *Not Ready* status.

Note

Refer to the *EMC Solutions Enabler Installation Guide* for license information required for restore functionality.

Options

-file
Use this with `FileName` to specify the devices associated with the session.

-session name
Use this with `SessionName` to specify the session to restore

-pace
Sets the speed of the session. See Set the ORS session pace on page 667

Example
To restore original data from a differential push session back to the control device, enter:

```
symrcopy restore -file dev_file_3
```

Restore ORS data using donor update

Description
Differential push operations are 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.

Options

- force
Terminates a session that has `-donor_update` enabled.
Turns off the `-donor_update` option while session is in *RestInProg*. The session will continue to restore in its current mode without donor update.
Use 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.

donor_update off -consistent
Maintains consistency on the remote devices after the restore session is complete. Must be set before restore action.

Examples
To restore data back to the control device using the `-donor_update` option, enter:

```
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.

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
```

Restrictions:

- If `-donor_update` option is used, a session cannot be renamed or devices cannot be removed from a session in the *Restored* state.
- 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.

VDEVs in Open Replicator

Open Replicator supports copy operations for a cold push session from a virtual device (VDEV). Open Replicator works with TimeFinder/Snap operations to copy data from an existing Snap session. This feature provides an alternative to hot copying, which requires all mapped directors must participate in the session.

To start this process, 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. In a cold push session from a VDEV, the VDEV must be mapped to at least one director, and the director must be zoned to the remote device. Only the directors mapped to the VDEV are required to adhere to the zoning rules for Open Replicator. Directors mapped to standard devices are not required to be zoned to the remote device.

Device requirements

The following devices and configuration requirements are required 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.

Operating system restrictions

The following operating system restrictions apply to Open Replicator sessions when using VDEVs:

- Supported only for arrays running Enginuity version 5874 through 5876. Refer to the *EMC Solutions Enabler Installation Guide* for license information on Snap devices.
- For arrays running HYPERMAX OS 5977 or higher, creating ORS sessions on control devices that are involved in a TimeFinder session is not supported.

Perform an ORS session with VDEVs

The following steps explain how to run an Open Replicator session with VDEVs using TimeFinder/Snap and Open Replicator:

Procedure

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.

Recreate ORS session with VDEVS

The following steps describe how to recreate, activate, and terminate the sessions: Example 5: Perform a cold push from a VDEV on page 704 provides a detailed example.

Procedure

1. Recreate the TimeFinder/Snap session using the symsnap **`re`** create command. This command clears the VDEV pointers and sets the Open Replicator protected tracks.
2. Recreate the Open Replicator session using the symrcopy **`re`** create 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.

Open Replicator examples

This section provides the following examples of using Open Replicator:

Example 1: Perform an ORS hot pull operation on page 680

Example 2: Pushing data using BCVs as the control devices on page 685

Example 3: Pull data online from IBM F20 to VMAX array on page 691

Example 4: Pushing online data from VMAX array to HDS 9960 array on page 698

Example 5: Perform a cold push from a VDEV on page 704

Example 1: Perform an ORS hot pull operation

This example shows how to migrate data from an older array to a VMAX array. The hardware setup consists of the control array with array ID 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 steps for performing a hot pull operation are:

- Example 1 Hot pull step 1 – create device file on page 681
- Example 1 hot pull step 2 – create migration session with donor update on page 681
- Example 1 hot pull step 3 – query migration session on page 681
- Example 1 hot pull step 4 – activate migration session on page 682
- Example 1 hot pull step 5 – verify session status on page 683
- Example 1 hot pull step 6 – terminate migration session on page 685

Example 1 Hot pull step 1 – create 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=60060480000000003143535 94D303737
symdev=000187900041:EA  wwn=60060480000000003143535 94D303738
symdev=000187900041:EB  wwn=60060480000000003143535 94D303739
```

Example 1 hot pull step 2 – create 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'.
```

Example 1 hot pull step 3 – query 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 query` 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.

Example 1 hot pull step 4 – activate 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 query` 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.

Example 1 hot pull step 5 – verify 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.
```

Example 1 hot pull step 6 – terminate 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'.
```

Aother `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.

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 steps for pushing data using BCVs as control devices are:

- Example 2 pull operation with BVC control devices step 1 create device group on page 686
- Example 2 pull operation with BVC control devices step 2 – create device file on page 686
- Example 2 pull operation with BVC control devices step 3 – establish TimeFinder session on page 686
- Example 2 pull operation with BVC control devices step 4 – split TimeFinder session on page 687
- Example 2 pull operation with BVC control devices step 5 – create migration session on page 687
- Example 2 pull operation with BVC control devices step 6 – activate migration session on page 688

- Example 2 pull operation with BVC control devices step 7 – query migration session on page 688
- Example 2 pull operation with BVC control devices step 8 – Check for invalid tracks on standard device on page 689
- Example 2 pull operation with BVC control devices step 9 – re-establishing TimeFinder session on page 689
- Example 2 pull operation with BVC control devices step 10 – recreate migration session for incremental update on page 690
- Example 2 pull operation with BVC control devices step 11 – verify session status on page 691

Note

For arrays running HYPERMAX OS 5977, push operations are not supported.

Example 2 push operation with BCV control devices – 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**), and two remote devices on a remote array (**sid 60**).

Example 2 pull operation with BVC control devices step 1 create 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
```

Example 2 pull operation with BVC control devices step 2 – create 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 with LUN WWN `60060480000184500160535 94D374646`.

```
vi conga
symdev=35:B8  wwn=60060480000184500160535 94D374646
symdev=35:B9  wwn=60060480000184500160535 94D383030
```

Example 2 pull operation with BVC control devices step 3 – establish 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.
```

Example 2 pull operation with BVC control devices step 4 – split 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.
```

Example 2 pull operation with BVC control devices step 5 – create 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'.
```

Example 2 pull operation with BVC control devices step 6 – activate 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.
```

Example 2 pull operation with BVC control devices step 7 – query 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.
```

Example 2 pull operation with BVC control devices step 8 – Check 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.
```

Example 2 pull operation with BVC control devices step 9 – 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.
```

Example 2 pull operation with BVC control devices step 10 – recreate 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'.
```

Example 2 pull operation with BVC control devices step 11 – verify 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: Pull data online from IBM F20 to VMAX array

This example shows how to migrate using a hot pull from an IBM F20 array to a VMAX 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 array, and how to perform operations on the remote host connected to the F20 array.

Steps for pulling online data from IBM F20 to VMAX array are:

1. Example 3 pull data online from IBM F20 to VMAX array step 1 – identify IBM devices on page 692 — gets WWNs of the IBM array devices.
2. Example 3 pull data online from IBM F20 to VMAX array step 2 – create device file on page 692
3. Example 3 pull data online from IBM F20 to VMAX array step 3 – create migration session on page 692
4. Example 3 pull data online from IBM F20 to VMAX array step 4 – shutdown remote application running on IBM F20 devices on page 693
5. Example 3 pull data online from IBM F20 to VMAX array step 5 – activate migration session on page 693
6. Example 3 pull data online from IBM F20 to VMAX array step 6 – query migration session on page 693
7. Example 3 pull data online from IBM F20 to VMAX array step 7– set ceiling value on page 694
8. Example 3 pull data online from IBM F20 to VMAX array step 8 – adjusting ceiling value on page 695

NOTICE

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.

Example 3 pull data online from IBM F20 to VMAX 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.

Example 3 pull data online from IBM F20 to VMAX array step 1 – identify 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 49424d2020202020323130352020202020202020202020303237323034393
/dev/rdsk/c3t10d1s2 02820499 49424d2020202020323130352020202020202020202020303238323034393
/dev/rdsk/c3t10d2s2 02920499 49424d2020202020323130352020202020202020202020303239323034393
/dev/rdsk/c3t10d3s2 02A20499 49424d2020202020323130352020202020202020202020303241323034393
/dev/rdsk/c3t10d4s2 02B20499 49424d2020202020323130352020202020202020202020303242323034393
/dev/rdsk/c3t10d5s2 02C20499 49424d2020202020323130352020202020202020202020303243323034393
/dev/rdsk/c3t10d6s2 02D20499 49424d2020202020323130352020202020202020202020303244323034393
/dev/rdsk/c3t10d7s2 02E20499 49424d2020202020323130352020202020202020202020303245323034393
/dev/rdsk/c3t10d8s2 02F20499 49424d2020202020323130352020202020202020202020303246323034393
/dev/rdsk/c3t10d9s2 03020499 49424d2020202020323130352020202020202020202020303330323034393
```

Example 3 pull data online from IBM F20 to VMAX array step 2 – create 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 `49424d2020202020323130352020202020202020202020303237323034393 9`. The next control device is paired with the next remote IBM device, and so forth:

```
vi devfile.pull
SYMDEV=25:1C4  wwn=49424d2020202020323130352020202020202020202020303237323034393
SYMDEV=25:1C5  wwn=49424d2020202020323130352020202020202020202020303238323034393
SYMDEV=25:1C6  wwn=49424d2020202020323130352020202020202020202020303239323034393
SYMDEV=25:1C7  wwn=49424d2020202020323130352020202020202020202020303241323034393
SYMDEV=25:1C8  wwn=49424d2020202020323130352020202020202020202020303242323034393
SYMDEV=25:1C9  wwn=49424d2020202020323130352020202020202020202020303243323034393
SYMDEV=25:1CA  wwn=49424d2020202020323130352020202020202020202020303244323034393
SYMDEV=25:1CB  wwn=49424d2020202020323130352020202020202020202020303245323034393
SYMDEV=25:1CC  wwn=49424d2020202020323130352020202020202020202020303246323034393
SYMDEV=25:1CD  wwn=49424d2020202020323130352020202020202020202020303330323034393
```

Example 3 pull data online from IBM F20 to VMAX array step 3 – create 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'.
```

Example 3 pull data online from IBM F20 to VMAX array step 4 – shutdown remote application running on IBM 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` operation is successful and the setup is correct before incurring application down time.

Example 3 pull data online from IBM F20 to VMAX array step 5 – activate 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` operation 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.

Example 3 pull data online from IBM F20 to VMAX array step 6 – query 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 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        0    5 IBM
000187990125:001C5     304382         0 49424D2020202020323130352020202* .W X..X.S.
```

```
CopyInProg         0     5 IBM
000187990125:001C6      304383          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001C7      304384          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001C8      304387          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001C9      304387          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001CA      304389          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001CB      304390          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001CC      304391          0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg         0     5 IBM
000187990125:001CD      304391          0 49424D2020202020323130352020202* .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.

Example 3 pull data online from IBM F20 to VMAX array step 7– set 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
```

Example 3 pull data online from IBM F20 to VMAX array step 8 – 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 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001C5     299415         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001C6     299500         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001C7     299584         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001C8     299692         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001C9     299772         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001CA     299774         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001CB     299989         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001CC     300146         0 49424D2020202020323130352020202* .W X..X.S.
CopyInProg        1    5 IBM
000187990125:001CD     301044         0 49424D2020202020323130352020202* .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 online data from VMAX array to 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 steps for pushing online data from VMAX array to HDS 9960 array :

1. Example 4 push online data from VMAX array to HDS 9960 array step 1 – identify HDS devices on page 698 — gets WWNs of the HDS array devices.
2. Example 4 push online data from VMAX array to HDS 9960 array step 2 – create device file on page 699
3. Example 4 push online data from VMAX array to HDS 9960 array step 3– create migration session on page 699
4. Example 4 push online data from VMAX array to HDS 9960 array step 4 – set ceiling value on page 700
5. Example 4 push online data from VMAX array to HDS 9960 array step 5 – activate migration session on page 701
6. Example 4 push online data from VMAX array to HDS 9960 array step 6– adjust ceiling value on page 702

Note

For arrays running HYPERMAX OS 5977, push operations are not supported.

NOTICE

Applications against remote HDS devices cannot be run at any time during the copy operation.

Example 4 push online data from VMAX array to HDS 9960 array – 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 array to the HDS array.
2. Configure the HDS array to assign devices to the VMAX array.

Example 4 push online data from VMAX array to HDS 9960 array step 1 – identify 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            484954414348492052343030394342443030303030  R400
/dev/rdsk/c3t10d1s2       65535            484954414348492052343030394342443030303032  R400
/dev/rdsk/c3t10d2s2       65535            484954414348492052343030394342443030303034  R400
/dev/rdsk/c3t10d3s2       65535            484954414348492052343030394342443030303036  R400
/dev/rdsk/c3t10d4s2       65535            484954414348492052343030394342443030303038  R400
/dev/rdsk/c3t10d5s2       65535            484954414348492052343030394342443030303041  R400
/dev/rdsk/c3t10d6s2       65535            484954414348492052343030394342443030303043  R400
/dev/rdsk/c3t10d7s2       65535            484954414348492052343030394342443030303045  R400
```

Example 4 push online data from VMAX array to HDS 9960 array step 2 – create 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 `484954414348492052343030394342443030303030`. The next control device is paired with the next remote HDS device, and so forth:

```
vi devfile.push
symdev=25:1C4  wwn=484954414348492052343030394342443030303030
symdev=25:1C5  wwn=484954414348492052343030394342443030303032
symdev=25:1C6  wwn=484954414348492052343030394342443030303034
symdev=25:1C7  wwn=484954414348492052343030394342443030303036
symdev=25:1C8  wwn=484954414348492052343030394342443030303038
symdev=25:1C9  wwn=484954414348492052343030394342443030303041
symdev=25:1CA  wwn=484954414348492052343030394342443030303043
symdev=25:1CB  wwn=484954414348492052343030394342443030303045
symdev=25:1CC  wwn=484954414348492052343030394342443030303130
symdev=25:1CD  wwn=484954414348492052343030394342443030303132
```

Example 4 push online data from VMAX array to HDS 9960 array step 3– create 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.push` 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 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001C5     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001C6     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001C7     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001C8     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001C9     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001CA     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001CB     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001CC     435150 484954414348492052343030394342 4* .W XXXX.S. Created         N/A
000187990125:001CD     435150 484954414348492052343030394342 4* .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.
```

Example 4 push online data from VMAX array to HDS 9960 array step 4 – set 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
```

Example 4 push online data from VMAX array to HDS 9960 array step 5 – activate 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'.
```

Example 4 push online data from VMAX array to HDS 9960 array step 6– adjust 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: Perform 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. Example 5 performing cold push from VDEV step 1 – create TimeFinder/Snap session on page 705
2. Example 5 performing cold push from VDEV step 2 – create migration session on page 706
3. Example 5 performing cold push from VDEV step 4 – activate migration session on page 707
4. Example 5 performing cold push from VDEV step 4 – query migration session on page 708

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.

Example 5 performing cold push from VDEV step 1 – create TimeFinder/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.
```

Example 5 performing cold push from VDEV step 2 – create 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.
```

Example 5 performing cold push from VDEV step 3 – activate TimeFinder/Snap session

Activate the TimeFinder/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.
```

Example 5 performing cold push from VDEV step 4 – activate 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'.
```

Example 5 performing cold push from VDEV step 4 – query 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 710
- ORS operations allowed for replication session states 710
- ORS operations allowed for device types 723

Device control operations allowed for ORS control devices

Table 58 on page 710 details if a device control operation is permissible, for the ORS control device, in various states with the controlling host (such as, ready, not ready).

Table 58 Device control operations allowed for ORS control device states with host

Action	Allowed
Ready	
Online RCopy control device pushing out	Yes
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
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
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
Online RCopy control device pulling in	Yes
Offline RCopy control device pushing out	No
Offline RCopy control device pulling in	No

ORS operations allowed for replication session states

Use the state tables to determine if an ORS operation is permissible on devices in various states of a replication (TimeFinder or SRDF) session:

- TimeFinder/Mirror and TimeFinder/Clone sessions — Details if an ORS operation is supported on devices in various states of a TimeFinder session.
- TimeFinder/SnapVX sessions (HYPERMAX OS 5977) — Details if an ORS operation is supported on devices in various states of a TimeFinder/SnapVX session.
- SRDF sessions — Details if an ORS operation is supported on devices (such as R1 and R2) in various states of SRDF sessions.

TimeFinder/Mirror and TimeFinder/Clone sessions

The following tables detail which ORS (rcopy) operations allowed when the control device is in use as a source or target for TimeFinder/Mirror, or as a source or target for TimeFinder/Clone:

Table 59 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Mirror SOURCE

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in prog	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Activate	Y	Y	Y	Y	Y	Y	Y	Y		Y		
Create	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]		Y[a]		
Incremental Restore	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y
Recreate	Y	Y	Y	Y	Y	Y	Y	Y		Y		
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Precopy	Y	Y	Y	Y	Y	Y	Y	Y		Y		
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 60 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Mirror TARGET

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in progress	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Activate	Y						Y	Y				
Create	Y[a]						Y[a]	Y[a]				
Incremental Restore	Y					Y	Y	Y			Y	
Recreate	Y						Y	Y				
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Precopy	Y						Y	Y				
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 61 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror SOURCE

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in progress	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Activate	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	
Create	Y	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]		Y[a]	Y[a]	
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 61 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror SOURCE (continued)

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in progress	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 62 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror TARGET

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in progress	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Activate	Y					Y	Y	Y			Y	
Create	Y					Y[a]	Y[a]	Y[a]			Y[a]	
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 62 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Mirror TARGET (continued)

rcopy Operation	BCV Pair State											
	Never Established	Sync in prog	Syncronized	Split in progress	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 63 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Clone SOURCE

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Activate	Y[a]	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Ya	Yab	Ya	Ya	Ya
Create	Y[a]	Yabc	Yabc	Yabc	Yabc	Yabc	Yabc	Yabc	Yabc	Yabc	Yac	Yabc	Yac	Yac	Yac
Incremental Restore	Y[a]	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd		Yabd			
Recreate	Y[a]	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Ya	Ya	Ya	Ya	Ya
Remove	Y[a]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y[a]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SetPrecopy	Ya	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Yab	Ya	Ya	Ya	Ya	Ya
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 63 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Clone SOURCE (continued)

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 64 Allowable rcopy operations when the control device for PUSH session is in use as a TimeFinder/Clone TARGET

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Activate	Y[a]						Y[abd]			Y[abd]					
Create	Y[a]						Y[abcd]			Y[abcd]					
Incremental Restore	Y[a]						Y[abd]			Y[abd]					
Recreate	Y[a]						Y[abd]			Y[abd]					
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SetPrecopy	Y[a]						Y[abd]			Y[abd]					
Terminate	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 65 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone TARGET

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Activate	Y[a]						Y[abd]			Y[abd]					

Table 65 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone TARGET (continued)

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Create	Y[a]						Yabcd			Yabcd					
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	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 66 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone SOURCE

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Activate	Y[a]	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd	Yabd		Yabd		Yabcd			
Create	Y[a]	Yabcd	Yabcd	Yabcd	Yabcd	Yabcd	Yabcd	Yabcd	Yabcd	Yabcd		Yabd			
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 66 Allowable rcopy operations when the control device for PULL session is in use as a TimeFinder/Clone SOURCE (continued)

rcopy Operation	Pair State														
	No session	Create in prog	Created	Recreated	Precopy	Copy in progress	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	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.

Snap VX sessions

The following tables detail which ORS (rcopy) operations are allowed when the rcopy control device is in use as either a Snap VX source device or target device:

Table 67 Allowable rcopy operations when the control device for PUSH session is in use as a Snap VX SOURCE

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Activate										
Create										
Incremental Restore										
Recreate										
Remove										
Rename										
Set Copy										
Set Nocopy										
Set Pace										

Table 67 Allowable rcopy operations when the control device for PUSH session is in use as a Snap VX SOURCE (continued)

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Set Precopy										
Terminate	Y	Y	Y	Y	Y	Y	Y			

Table 68 Allowable rcopy operations when the control device for PUSH session is in use as Snap VX TARGET

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Activate										
Create										
Incremental Restore										
Recreate										
Remove										
Rename										
Set Copy										
Set Nocopy										
Set Pace										
Set Precopy										
Terminate	Y						Y	Y	Y	

Table 69 Allowable rcopy operations when the control device for PULL session is in use as a Snap VX SOURCE

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Activate										
Create										
Incremental Restore										
Recreate										
Remove										
Rename										
Set Copy										
Set Nocopy										
Set Pace										
Set Precopy										
Terminate	Y	Y	Y	Y	Y	Y	Y			

Table 70 Allowable rcopy operations when the control device for PULL session is in use as a Snap VX TARGET

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Activate										
Create										
Donor Update Off										
Failback										
FrontEnd Zero Off										
Remove										
Rename										

Table 70 Allowable rcopy operations when the control device for PULL session is in use as a Snap VX TARGET (continued)

rcopy Operation	TimeFinder/Snap VX State									
	No session	Establish in progress	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in progress	Link copied	Linked
Set Copy										
Set Nocopy										
Set Pace										
Terminate	Y						Y	Y	Y	

SRDF sessions

The following tables detail which ORS (rcopy) operations are allowed when the rcopy control device is in use as a SRDF R1 or R2 mirror:

Table 71 Allowable rcopy operations when the control device for PUSH session is in use as an SRDF R1 mirror

rcopy Operation	SRDF State											
	Sync in prog	Syncronized	Split	Suspended	Failed over	Partitioned1	Partitioned2	R1 updated	R1 updinprog	Invalid	Consistent	TransmitIdle
Activate	Y[b]	Y	Y	Y		Y	Y				Y[b]	Y[b]
Create	Y[ab]	Y[a]	Y[a]	Y[a]		Y[a]	Y[a]				Y[ab]	Y[ab]
Incremental Restore	Y[bc]	Y	Y	Y		Y	Y				Y[bc]	
Recreate	Y[b]	Y	Y	Y		Y	Y				Y[b]	Y[b]
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Precopy	Y[b]	Y	Y	Y		Y	Y				Y[b]	Y[b]
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 71 Allowable rcopy operations when the control device for PUSH session is in use as an SRDF R1 mirror (continued)

b. Must have no local invalids on R1 or remote invalids on R2.
c. Array must be running Enginuity 5874 or higher.

Table 72 Allowable rcopy operations when the control device for PUSH session is in use as an SRDF R2 mirror

	SRDF State											
rcopy Operation	Sync in prog	Syncronized	Split	Suspended	Failed over	Partitioned1	Partitioned2	R1 updated	R1 updinprog	Invalid	Consistent	TransmitIdle
Activate			Y	Y	Y	Y	Y	Y	Y			
Create			Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]			
Incremental Restore			Y	Y	Y	Y	Y	Y[b]				
Recreate			Y	Y	Y	Y	Y	Y	Y			
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Precopy			Y	Y	Y	Y	Y	Y	Y			
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.
b. R2 must be write enabled.

Table 73 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R1 mirror

	SRDF State											
rcopy Operation	Sync in prog	Syncronized	Split	Suspended	Failed over	Partitioned1	Partitioned2	R1 updated	R1 updinprog	Invalid	Consistent	TransmitIdle
Activate	Y[abcd]	Y[abcd]	Y	Y		Y	Y				Y[abcd]	
Create	Y[abcde]	Y[abcde]	Y[e]	Y[e]		Y[e]	Y[e]				Y[bcde]	
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Frontend Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 73 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R1 mirror (continued)

rcopy Operation	SRDF State											
	Sync in prog	Syncronized	Split	Suspended	Failed over	Partitioned1	Partitioned2	R1 updated	R1 updinprog	Invalid	Consistent	TransmitIdle
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Array must be running 5874 or higher.
b. Must have no local invalids on R1 or remote invalids on R2.
c. Frontend zero must be in Off state.
d. Force flag must be set.
e. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 74 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R2 mirror

rcopy Operation	SRDF State											
	Sync in prog	Syncronized	Split	Suspended	Failed over	Partitioned1	Partitioned2	R1 updated	R1 updinprog	Invalid	Consistent	TransmitIdle
Activate			Y		Y	Y	Y	Y[bc]				
Create			Y[a]		Y[a]	Y[a]	Y[a]	Y[abc]				
Frontend Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Failback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Donor Update Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Copy	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Remove	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.
b. R2 must be write enabled.

Table 74 Allowable rcopy operations when the control device for PULL session is in use as an SRDF R2 mirror (continued)

c. Frontend zero detect must be in Off state.

ORS operations allowed for device types

Table 75 on page 723 details if 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, see Virtual Provisioning overview on page 332.

Table 75 ORS operations allowed by device type

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 and 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 and 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 and CKD_3390	No
Rcopy Create pull, where remote device type is:	

Table 75 ORS operations allowed by device type (continued)

WORM	No
CKD_3380 and 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 726
- The FLM process 727
- Migration using FLM 729

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.

NOTICE

FLM is not supported on arrays running HYPERMAX OS 5977.

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.

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.*

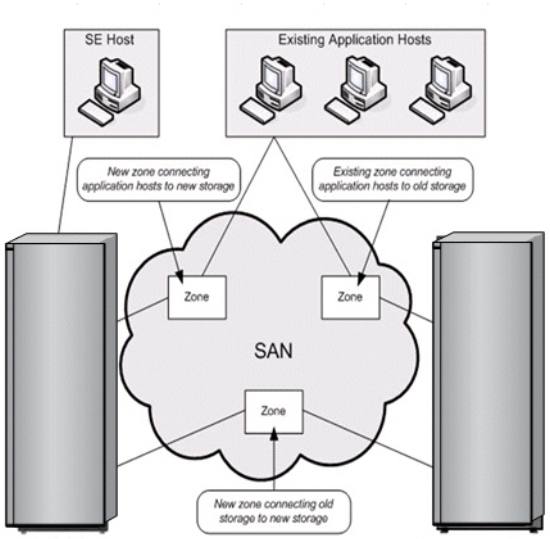
NOTICE

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 multipath type.

All documents mentioned above are available at http://support.EMC.com.

An FLM configuration including the network, storage arrays, application hosts, and the Solutions Enabler host, along with the required zoning and masking, is shown in the figure below.

Figure 26 Example FLM configuration



The FLM process

The FLM process steps are:

1. Verify that the donor array is zoned to the application host.
2. Zone the donor array to the new array.
3. Zone the new array to the application host.
4. Create the FLM session using a device file containing pair devices to copy 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.
5. Activate the FLM session, and the donor array devices are put into host passive mode and the new array device into host_active mode.
6. The host sees a new path for the donor devices, and 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.

FLM array/device requirements

FLM requires the following Enginuity versions and array/device configuration:

- Enginuity 5876 on the new arrays.
- Enginuity 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.

FLM 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 for FLM

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 361.

Configure the application hosts for FLM

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.

FLM prerequisites for zoning and masking

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:

Note

Due to requirements of multipath environments, paths must be configured to the new array devices to maintain data availability. Refer to SAN setup requirements on page 650 for more information on zoning requirements. For more information on zoning and masking refer to Masking devices with auto-provisioning groups on page 377. 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).

FLM prerequisites for zoning and masking

FLM prerequisites for zoning and masking

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.

NOTICE

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 650 for more information on zoning requirements. For more information on zoning and masking refer to Masking devices with auto-provisioning groups on page 377. Verify the application donor array is visible on the application host and error-free.

Prior to performing a FLM, zoning and masking must be configured as follows:

- 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).

Migration using FLM

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

Create a device file for FLM

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
```

Create an FLM session

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, otherwise the geometry is changed by the `symrcopy create` command. While the migration session is active, the geometry cannot be changed using the `symconfigure` command.

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.

Options

-migrate
Specifies that the session is an FLM session. This option must be used on all subsequent FLM operations.

-f
Specifies the file name.

-pull
Specifies that the data is copied from the donor array to the new array.

-host_type
Specifies the host operating system.

-hba_type
Specifies the HBA type.

-mp_type
Specifies multi-pathing type.

Example

To create a migration session with specific host, HBA, and multi--pathing type, enter:

```
symrcopy -f win_flm create -pull -migrate -host_type windows -
hba_type Qlogic -mp_type PPath_45 -v
```

Restrictions

- For the `symrcopy create` command, `-mp_type`, `-host_type`, and, `-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.

- 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 667 for more information on the `set pace` command.
- If the session creation fails at any point, the system is left in the same state prior to the `symrcopy create` command.

Verify the created FLM session

Example

To verify that the FLM paris are listed as migration sessions in the *Created* state, enter:

```
symrcopy -f win_flm query
```

Sample output

```
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.
```

FLM – restore new array device to host-active mode

Syntax

If a created session is terminated before it is activated, the new array device can be restored to host-active mode. To restore a new array device to host-active mode, use the following syntax:

```
symdev host_active -sid SymmID SymDevName
```

FLM session activation

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.

Note

Do not proceed with the `activate` command unless all the new array devices are configured as new paths to each of the donor array devices involved in FLM. The next step in the process sets 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 is rejected. For details on performing these steps, refer to *EMC Symmetrix Procedure Generator* available at https://support.EMC.com.

Activate an FLM session

Options

-symforce
Forces the activate operation to execute when it would otherwise be rejected. Use caution when using this option.

Examples

To activate a migration session, enter:

```
symrcopy activate -f win_flm -migrate
```

Note

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 remains 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 sessions on the donor array device are not migrated.

Set the FLM ceiling value

Description

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 40%:

Note

Ceiling values may need to be adjusted to optimize the performance of the specific SAN environment.

Examples

To set a bandwidth ceiling of 40% for all directors on array `307`, enter:

```
symqos -rcopy set ceiling 40 -dir all -sid 307
```

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 667.

List the FLM ceiling setting

Example

To list the current ceiling value, enter:

```
symqos -rcopy list ceiling
```

Monitoring FLM session status

An FLM session status is checked using the `symrcopy query`, `symrcopy verify`, or `symrcopy list` command.

Query FLM session status

Examples

To query FLM session status

```
symrcopy query -f win_flm
```

To query FLM status using the `-summary` option, enter:

```
symrcopy query -f win_flm -summary
```

Sample output

For session status 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. 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

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.
```

For the

`summary query`

command, an abbreviated listing shows all possible session states and the number of sessions that are in each state:

```
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
```

Verify FLM session status

Examples

To verify session status, enter:

```
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 672 for more information on verify options.

To verify session status with the `-summary` option, enter:

```
symrcopy -f win_flm verify -summary
```

Sample output

Note

The one-line `verify` command displays at the bottom of 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 672 for more information on verify options.

List FLM session

Description

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.

Examples

To list FLM sessions, enter:

```
symrcopy list -sid 307 -type migrate
```

Sample output

```
Symmetrix ID: 000194900307
Control Device             Remote Device              Flags      Status     Done
---------------------- ---------------------------------- ------ -------------- ----
          Protected
Sym       Tracks                    Identification               RI CDSHUTZ    CTL <=> REM
(%)
-----   ------------ ---------------------------------- -- ------ -------------- ----
00171             30000        600604800000000619053594D314638 SW ...XXM.      Copied
N/A
00172             30000        600604800000000619053594D314637 SW ...XXM.      Copied
N/A
00173             30000        600604800000000619053594D314642 SW ...XXM.      Copied
N/A
00174             30000        600604800000000619053594D314646 SW ...XXM.      Copied
N/A
Total ---------
  Track(s) 120000
  MB(s)      80062.5

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.
```

Note

For listing all session types or only listing RecoverPoint sessions, refer to Listing all sessions on page 668 and Filtering session types on page 669.

FLM session fallback

Session failback can occur automatically or it can be performed manually. The following section describes automatic and manual failback of an FLM session.

Automatic FLM session 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 FLM session 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.

Failback of manual FLM session

Examples
To failback an FLM session, enter:

```
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.

Terminate FLM session

Description
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.

Examples

To terminate FLM session `sym1_to_sym2`, enter:

```
symrcopy terminate -session_name sym1_to_sym2 -migrate
```

FLM – restore 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.

Note

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.

Restore donor device to host-active mode

Description

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:

- 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.

Syntax

To restore donor device to `host_active` mode, use the following syntax:

```
symdev host_active -sid SymmID SymDevName
```

INDEX

A

Abort configuration session 227
access control backup file 81
access control configuration 62
access control database 60
access control for device masking 413
access control IDs 61
access control operations 62
access control permissions 70
access control strategies 76
access group permissions 70
access groups 64
ACL setup 63
activate devices in thin pool 350
activate edisk 374
activate flm session 732
activate ORS session 665
activating VCMDB 426
add child storage groups to storage group 105
add devices to access pool 68
add devices to cache partitions 462
add devices to device group 88
add devices to storage group 103, 104
add devices to storage groups 388
add devices to target list 89
add disk group to FAST tiers 185
add edisk 368
add host access id to group 64
add initiators 393
add initiators to initiator group 393
add initiators using input file 393
add IP route 318
add masked devices 428
add masked devices from device group 429
add masked devices using device file 430
add masked devices with dynamic LUN address 430
add masked reserved devices 429
add members to concatenated members 256
add members to striped meta 257
add ports to port groups 390
add thin pools to FAST tier 187
add tier to FAST policy 193
add time wndows 609
add ungrouped devices 90
add user access id to group 65
add virtual devices to device group 89
adding edisk restrictions 368
advise, reservations 274
allocate space thin devices 345
analyze mode cache partitions 464
approve FAST plan 214
array external locks 507
array-level data 492
associate storage group to policy 198
attach ip interface to iscsi target 315, 316
audience 23
audit log
    viewing 484
audit log for single array 633
audit log for specified time period 634
authentication types 50
authorization rules 49
auto-provisioning groups 378
automated pool rebalancing 353

B

backend and prefetch reporting 559
background operations on thin devices 347
backup and restore authentication information 418
backup and restore masking view 386
backup authentication database 418
backup masking database 435
backup the authorization database 55
bind thin devices 342
bind thin metadevice to pool 260
block writes to vdmdb 436
Bring front-end directors online 506
Bring RA directors online 506
broken sequences in skip adjustment 445

C

cache partition 462
cache statistics 564
cascaded groups 109
cascaded initiator groups 394
cascaded storage group details
    grouping 121
cascaded storage groups 111
ceiling value setting 666
change device state 526
change device state using file 526
change host I/O limits 116
change storage group workload type 108
change tracker command summary 573
change tracker device file 575
change tracker overview 572
change tracker reports 579
changing database file 42
chap authentication 415
check free physical disk space 228
CKD devices
    assigning PAV aliases 284
    copying a CU image 284
Clerra devices 91
cold copy 661
command line help 30
comments 23
compatibility 33
compatibility restrictions 33

composite groups 132
configuration change 224
Configuration change guidelines 227
configuration change session rules 224
configuration data overview 488
configuration data synch 43
configuration flags 448
configure iscsi two-way CHAP 423
configure iscsi with CHAP 420
configure iscsi without CHAP 423
connectivity authorization 39
conventions for publication 23
convert a thin device 263
convert cascaded group to storage group 112
convert devices 260
convert storage group to cascaded group 111
convert vcmdb types 435
copy all devices between storage groups 129
copy cu image mapping 284
copy device between storage groups 128
copy device range mapping 284
copy devices between device groups 97
copy devices from device group to storage group 97
copy groups and masking views 387
copy pace 452
copy pacing 454
copy port groups 392
create access group 64
create access pool 68
Create alternate ACCID 78
create and manage ACEs 69
create AS/400 metadevices 255
create auto-provisioning session 380
create cache partitions 461
create change tracker reports 581
create change tracker session 575
create DATA devices 334
create device file for FLM 730
create device group 87
create device restrictions 239, 247
create devices 236, 240, 242
Create empty device group 87
create external disk group 366
create FAST disk group tiers 183
create FAST policy 192
create FAST VP tiers 184
create flm session 730
create groups and views 382
create initiator groups 383
create IP interface 305
Create iSCSI target 309, 311
create masking view 379, 384
create or modify access control data 61
create ORS copy session 658
create port groups 383
Create RDF metadevices 254
create storage groups 101, 382
create thin devices 335
create thin devices with file 335
create thin pools 337

D

D@RE 496
DATA device rules 335
DATA devices
    draining 279
data preservation 258
database access modes 42
database configuration information 43
database file location 41
database file lock 42
database location client/server mode 42
database locking 411
database release lock 411
deactivate devices in thin pool 349
decline FAST plan 214
default acces control configuration 77
define endpoint IP address 623
delete access group 67
delete access pool 69
delete cache partition 464
delete CHAP 399
delete device group 99
delete external disk group 367
delete FAST policy 194
delete FAST tiers 187
delete HBA associations 441
delete initiator groups 394
delete IP interface 308
delete iscsi target 313
delete masking view 385
delete port groups 391
delete RADIUS 426
delete storage group 131
delete storage groups 389
device
    identifiers 270
device attribute values 265
device control operations allowed 710
Device conversion restrictions 262
device emulation 527
device external locks 544
device group control operations 99
device group definition 161
device group invalid warning 263
device group names 84
device group types 85
device groups
    adding devices to 135
    creating 135
    exporting 123
    RDF1 134
    RDF2 134
device information 656
device list by array id 489
device list capacity output 544
device list with pddevfile format 490
device list with wwn 489
device list without capacity 489
device lists 85
device masking 377
device masking controls 411
device masking overview 410

device migrations
using virtual LUN 470
device reservations 273
device restrictions 86
device types 519
device types Enginuity 5876 or lower 520
device types HYPERMAX OS 5977 520
device-level data 519
device-level statistics
statistics 568
devices
deleting 271
mapping 281
director port statistics 561
director statistics 560
director types 501
disable CHAP 399
disable donor update 664
disable enable RADIUS 425
Disable environment variables 32
disable FAST 204, 206
Disable front-end zero detection 663
disassociate storage group to policy 199
discover host hba 427
discover host HBAs 381
discover overview 38
discovery syntax 38
disk space
configuring into devices 249
disk statistics 563
disk-level data 545
display authentication 417
display auto-provisioning information 401
display CHAP 399
display devices with no assignments 405
Display environment variables 31
display FAST activity history 215
display FAST state 204
Display full group names 32
display masking views 402
display srdf/a cache statistics 564
display srdf/a statistics 558
display time wndows 612
display virtual disks 41
Dissolve a metadevice 258
donor update 663
drain and rebalance thin pools 348
draining DATA devices 279
dynamic cache partitions 460
dynamic RDF
devices, configuring 289
enabling 290

E

eDisk encapsulated 364
eDisk encapsulation examples 366
eDisk external provisioning 365
eDisk geometry limited restrictions 365
eDisks 364
edit and manage access pools 69
empty cache partitions 465
enable CHAP 399
enable CHAP on iscsi initiator 399
enable disable cache partitions 461
enable disable priority of service 456
enable disable RDF workload distribution 459
enable donor update 663
Enable environment variables 31
enable FAST 203
Enable front-end zero detection 662
enable iSCSI authentication 415
enable or disable vp compression 353
enable RADIUS for iscsi initiator 426
Enable user authorization 53
enforce, reservations 274
enforcement 53
Enhanced Virtual LUN Technology
overview of 470
Environmental data 512
environmental data output display 513
establish administrator 78
event daemon 635
events and logs overview 630, 633
export device list
grouping 123
export device lists 93
export groups to files 123
export ORS device list 658
external spindle devices 524
external spindle information 553

F

failback of manual FLM session 737
FAST 5874 5876 180
FAST and FAST VP comparison 181
FAST audit log 215
FAST compliance report 216
FAST compliance report output values 217
FAST configuration 181
FAST control parameters 207
FAST controller 202
FAST HYPERMAX 168
FAST information display 214
FAST Optimizer permissions 81
FAST plan output values 213
FAST reporting 171
FAST reports 216
FAST SRDF coordination 198
FAST state output values 205
FAST storage groups 196
FAST tech demand report descriptions 220
FAST technology demand report 218
FAST tier restrictions 182
FAST-related storage groups 196
FC to host interface 442
FCID format 444
FCID lockdown 392, 443, 444
fibre protocols 296
filter by AS400_GK 543
filter by DA, interface, disk, hypervolume 542
filter by DIF1 542
filter by director 540

filter by director port mapping 541
filter by FAST 542
filter by metadevice 541
filter by RAID information 542
filter by SRDF devices 541
filter by storage group 542
filter by technology type 542
find FCID 442
FLM session failback 737
flm zoning and masking 729
forcible gerneration of alternate access ID 61
form metadevice 254
free all written tracks 347
free written tracks 346
front-end zero detection 662
FTS overview 362
FTS overview Enginuity 5876 363
FTS overview HYPERMAX OS 5977 362
FTS Solutions Enabler support 363

G

General absolute control 79
geometry edisk encapsulated devices 365
GNS
    output 161
    active group list 152
    composite groups 152
    daemon 157
    SRDF option 152
GNS daemon
    controls 155
GNS, daemon options
    remote mirrors 159
grant base permissions 73
grant bcv and sdr permissions 74
grant permissions to unknown user 74
grant srdf permissions 74
group management iscsi target 324
group name services 84

H

hba alias names 427
hba ids and ascii format 428
HBA initiators 412
hba port flags 446
heterogeneous host configuration 447
host
    updating 287
host access 410
host access control overview 60
host access id 76
host access unsupported hosts platforms 411
host configuration database 38
host configuration database file 41
host connections 501
host connections to arrays 500
host i/o priority 456
Host I/O priority 455
hot copy 660
hot pull activate migration session 682
Hot pull create device file 681
hot pull create migration session with donor update 681
hot pull operation 680
hot pull query migration session 681
hot pull terminate migration session 685
hot pull verify session status 683

I

Identify components for device masking 413
import device lists 94
import or export device lists 93
inhibit database synchronization 44
initial device masking 412
initial setup 79
initialize authentication database 418
initialize masking database 433
initialize vcmdb database 415
initiator group flags 395
IPsec 616–618, 620–622, 624, 625, 627, 628
IPsec and EMC arrays 616
IPsec command 617
IPsec command file syntax 622
IPsec commit policy file 628
IPsec file scripting 621
IPsec IKE errors 621
IPsec IKE options 625
IPsec key management 617
IPsec list policies 618
IPsec list policy statistics 620
IPsec manual options 624
IPsec overview 616
IPsec policy file 628
IPsec policy syntax 622
IPsec preview policy syntax 627
IPsec proposal options 625
IPsec proposal set syntax 624
IPsec show policy detail 618
IPsec transform options 625
iscsi configuration example 320
iscsi driver configuration 419
iscsi target reporting 324
iscsi targets 304

L

latency time 589
LED door feature 494
limit access to access pools 67
Limit access to UnknwGrp group 78
lin-level statistics
    statistics 569
list access control 75
list actual disk capacity 547
list all array locks 507
list all device groups 92
list all LRU groups 509
list allocations across multiple pools 534
list and show storage groups 390
list array bays 494
list array component environmental data 513
list array environmental data 512
list array IP addresses 321
list array IP routes 323

list array iscsi targets 322
list array LED states 495
list array statistics 558
list arrays 493
list awwn logins 440
list backend statistics 559
list changed devices 537
list CHAP 328
list cu image details 512
list cu images 512
list cycle statistics
statistics 566
List data and save devices 531
list data encryption audit log 498
list data encryption status 497
list database backup file 419
list device assignments 328
list device emulations 528
list device identifiers 492
list device information 327, 522
list device mapping information 491
list device pools 515
list device statistics 558
list devices in device group 92
list devices mapped to directors 504
list director data 502
list director details 503
list director port data 504
list director type 503
list disk gaps 547
list disk group summary 548
list disk group summary by engine 549
list disk groups 547
list disk information 545
list disk statistics 563
list DMX connections 414
list enginuity patches 512
list environmental data example 514
list environmental data for service state 513
list events 632
list external locks 544
list external spindle state 371
list external spindles 524
list FA director mappings 415
list FAST DP tiers 190
list FAST policy 195
list FAST tiers 188
list FAST VP tiers 189
list feature registrations and usage data 518
list filtered ORS session types 669
list flm ceiling value 733
List FLM session 736
list front-end director statistics 564
list hba 327
list hba and director ports 414
list hba information 490
list host HBAs 381
list initiator group devices 406
list iscsi statistics 561
list lock details 507
list lock number details 507
list logins 329
list marked objects for change monitoring 576
list memory board information 511
list next available device address 505
list ORS sessions 668
list physical device names 414
list pinned devices 539, 540
list port flags 502
list prefetch activity 559
list priority of service 456
list RAID group information 525
list rcopy director port statistics 561
list rcopy statistics 559
list SAS drives 546
list SLO 171
list spare physical disks 550
list spindle ID information 523
list spindle information 551
list spindle information for disk group 553
list spindle information for physical disk 552
list spindle statistics 563
list SRP 173
list SRP for thin devices 179
list SRP verbose 174
list storage group demand 329
list storage group FAST associations 200
list storage groups 118
List storage groups 172
list symapi services 511
list thin bcv devices 534
list thin device tier allocations 533
list thin devices 532
list thin devices in pool 533
list time windows summary 613
list user roles 56
list virtualized devices 537
locked access control session 76
log file configuration options 630
logical device names 90
login history table 438
LRU parameters 457
LUN base/offset 445
LUN visibility 444

M

mainframe cu image 511
manage access group 66
manage default access 77
manage FAST policies 191
manage FAST tiers 182
manage HBA initiators 440
manage initiator groups 393
manage masking view 385
manage port groups 390
manage storage groups 387
manual FLM session failback 737
mapping to front-end ports 281
mark objects for change monitoring 576
masking database maintenance 433
metadevice
thinthin devices
meta 259

metadevices
configuration restrictions 255
converting 264
creating 253
converting 264
metahead 251
performance 252
size requirements 252
migrated devices 537
migration sessions
listing 480
managing 480
terminating 483
modified reporting command for 5977 401
modify cache partition 463
modify device capacity 240
modify FAST tier limit 193
modify FAST VP tiers 186
modify IP interface 306
modify role mappings 54
modify SRP 169
modify storage group 200
modify storage groups 105
monitor authorization 56
monitor events 632
monitor flm session status 733
monitor thin pools 355
move access id to group 66
move all devices between device groups 96
move all devices between storage groups 128
move device between storage groups 126
move devices between device groups 95
multi session consistency 136
multiple cores and prots 503
multiple user roles 49
multisession consistency (MSC) 135
MVS subsystem IDs 249

N

name groups and views 386
netcnfg file 510
nonpooled device permissions 79

O

object reporting 556
one-way chap authentication 416
online offline modes 43
Open Replicator 644
Optimized Read Miss mode 509
Optimizer
best swaps with 589
command filecommand file
parameters 594
copying volumes to DRVs 591
legal swaps 591
rollbacks 602
setting parameters for 593
setting user approved mode 602
starting 593
swap procedure 590
display FAST control parameters
FAST 210
FAST control parameters
parameters 207
performance metrics 588
ORS 644, 645, 647–649
ORS and host 645
ORS background copy 665
ORS background copying 666
ORS ceiling 667
ORS cold push example 654
ORS command summary 651
ORS copying limitation 649
ORS device file 658
ORS device guidelines 649
ORS differential copy 659
ORS FTS support 649
ORS hot pull example 653
ORS limitations 644
ORS operation 652, 655, 656
ORS operational limitations 648
ORS operational rules and limitations 647
ORS operations allowed for device types 723
ORS precopy 659
ORS SAN setup 650
ORS state tables for replication 710
ORS uses
ORS 644
output
command line time-saving 33
compatibility mode 33
truncated output display 33

P

PAV alias addresses
adding to mapped devices 284
Perform a cold push from a VDEV 704
Perform a cold push from a VDEV - activate migration session 707
Perform a cold push from a VDEV - activate snap session 707
Perform a cold push from a VDEV - create migration session 706
Perform a cold push from a VDEV - create snap session 705
perform ors with vdevs 679
pin devices 539
point-in-time copy 666
port characteristics
setting 296
port flags 296
port group information 392
port statistics 561
post drive replacement audit log 498
preface 23
Preset names and IDs 32
preview and commit user authorization 55
prior scripts
scripts 34
provisioning limits 378
pull data online from IBM F20 to VMAX 691

pull data online from IBM F20 to VMAX activate migration session 693
pull data online from IBM F20 to VMAX create device file 692
pull data online from IBM F20 to VMAX create migration session 692
pull data online from IBM F20 to VMAX ID IBM devices 692
pull data online from IBM F20 to VMAX prerequisite tasks 692
pull data online from IBM F20 to VMAX query migration session 693
pull data online from IBM F20 to VMAX set ceiling value 694
pull data online from IBM F20 to VMAX shutdown remote application 693
pull operation with BCV control devices create device group 686
pull operation with BVC control devices activate migration session 688
pull operation with BVC control devices check invalid tracks 689
pull operation with BVC control devices create device file 686
pull operation with BVC control devices create migration session 687
pull operation with BVC control devices establish timefinder session 686
pull operation with BVC control devices query migration session 688
pull operation with BVC control devices re-establilsh TimeFinder session 689
pull operation with BVC control devices recreate migration session incremental update 690
pull operation with BVC control devices split TimeFinder session 687
pull operation with BVC control devices verify migration session status 691
push data using BCVs 685
push online data from VMAX array to HDS 9960 array 698
push online data from VMAX array to HDS 9960 array- activate migration session 701
push online data from VMAX array to HDS 9960 array- adjust ceiling value 702
push online data from VMAX array to HDS 9960 array- create device file 699
push online data from VMAX array to HDS 9960 array- create migration session 699
push online data from VMAX array to HDS 9960 array- ID HDS devices 698
push online data from VMAX array to HDS 9960 array- prereqs 698
push online data from VMAX array to HDS 9960 array- set ceiling value 700

Q

QoS overview 452
QoS query 454
Quality of Service
    setting the copy pace level 453
query flm session status 733

R

RA QoS values 459
RADIUS server 425
rcopy concepts 645
rcopy director statistics 560
RDF capacity planning
    using Change Tracker 582
RDF consistency group 135
rdf statistics
    statistics 568
RDF workload distribution 460
rebind thin devices 343
reclaim allocations on thin devices 345
recover from failed session 676
RecoverPoint permissions 80
recreate ors session
    ORS 675
recreate ors with vdevs 680
refresh vcmdb 439
related documentation 23
release array lock 508
release device external locks 544
release locked access control session 76
remove access ID 66
remove all ACEs from access group 67
remove all time wndows 612
remove device from device group 98
Remove devices from access pool 69
remove devices from cache partition 464
remove devices from device group 98
remove devices from storage group 112
remove devices from storage groups 389
remove disk groups from FAST tier 186
remove edisk 370
remove FAST tier 193
remove IP route 319
remove marked objects for change monitoring 577
remove masked devices 431
remove masked devices from device group 432
remove masked devices using device file 433
remove masked meta devices 432
remove metamembers 257
remove permissions from access group 75
remove ports from port groups 391
remove remote device from session 675
remove SLO and SRP from storage group 108
remove SLO from storage group 107
remove SRP from storage group 108
remove storage group workload 108
remove thin pools from FAST tier 187
remove time wndows 610
Remove UnknwGrp 77
rename awwn 440
rename cache partition 463
rename device groups 95
rename FAST policy 194
rename FAST tiers 187
rename hba 399
rename hba ids 428
rename iscsi target 314
rename logical device 95
rename SRP 170

rename storage group 126
rename storage groups 389
rename thin pools 339
replace HBA 441
replacing hba 398
report flag details 301
report iscsi information 321
report port information for iscsi target 326
reservations
committing changes to 275
releasing 276
reset RDF director workload}
QoS 459
restore access control backup file 82
restore and convert vcmdb type 436
restore authentication database 419
restore donor device to host-active mode 738
restore new array device identity 738
restore new array device to host-active 731
restore session with donor update 678
restore the authorization database 56
restoring session 677
restrictions
for configuring metadevices 255
for deleting devices 271
Revert UnknwGrp to default 77
Role 48

S

scan for devices 39
SCSI device list 488
SCSI protocols 296
SCSI-level data 488
SDR
mapping 281
seek time 589
session rollback 380
set array attributes 234
set bandwidth 457
set change tracker session type 575
set CHAP credentials 399
set device attributes 265
set flm ceiling value 732
set HBA flags 396
set hba port flags 446
set heterogeneous host configuration 448
set initiator group override flag 395
set LUN base/offset skip adjustment 445
set LUN visibility 445
set online/offline state iscsi 321
set Optimized Read Miss mode 510
set ORS session pace 667
set pool attributes 339
set RADIUS server configuration 425
set RDF workload distribution 459
set SLO name 235
set SRP for storage group 107
set storage group SLO 106
Set user authorization 52
show access control 75
show clone state flags 529
show device details 528
show device group details 92
show disk geometry details 530
show edisk information 369
show FAST policy 196
show FAST tiers 191
show iscsi targets 326
show SLO 172
show SRP 176
Show storage group FAST associations 201
show thin pool details 535
show thin pool rebalancing 517
SLO demand 177
SMC Unisphere permissions 81
Solutions Enabler overview 30
SRDF device permissions 80
SRDF group attributes
setting 291
SRDF/A
assigning session priority 291
SRP reporting 177
start drain on data devices 350
start drain on edisk 372
start or stop vp compression 354
start pool rebalancing 351
statistic overview 556
statistics
BCV devices 562
standard devices 562
statistics by director number
statistics 569
statistics by RA group
statistics 568
statistics display options 557
stop background operations on thin devices 348
stop drain on data devices 350
stop edisk drain 373
stop i/o activity 228
stop pool rebalancing 352
storage group demand 178
storage group details
grouping 121
storage groups 109
store vcmdb 436
storgnsd, see also GNS, daemon options 157
striped data 252
support information 23
supported device masking topologies 412
swap RDF devices 289
Symaccess permissions 81
SYMCLI commands
symrcopy query 670
symdg 135, 152
symevent 632
symcli device reference 31
symcli help options 30
symconfigure restrictions 320
symmigrate
validating migration sessions 474
symoptmz
command features 592
symsan command

ORS 651
sync masking database 434

T

Take front-end directors offline 506
Take RA directors offline 506
terminate and restart ORS session 663
terminate FLM session 737
terminate recoverpoint session 675
terminate session 674
thin device actions 341
thin device allocate and reclaim space 344
thin device i/o activity 263
thin device rules 337
thin devices
converting 262
thin pool draining rules 348
thin pool rebalancing 351
thin pool rules 339
time windows definitions 606
time windows formats 607
time windows restrictions 607
time wndows conversion 608
time wndows conversion guidelines 607
time wndows overview 606
time-saving 33
timefinder permissions 79
TimeFinder/Snap permissions 80
transfer time 589
truncated 33
two-way chap authentication 417

U

udpate configuration data 44
unauthorized modification of VCMDB 434
unbind thin devices 342
unmap devices 286
unpin devices 539
Updating the host 287
user approved mode, for Optimizer 602
user authorization 48
User authorization 52
user authorization command file 54
user ID formats 50
username groupname formats 51

V

vcmbb 410
vcmdb database types 435
vdevs in ORS 678
verfity array configuration 228
verify admin authority 64, 68, 70
verify auto-provisioning database 400
verify device pool binding 344
verify device usge 537
verify edisk state 371
verify flm session 731
verify FLM session status 735
verify ORS session 672
verify pool and device states 355
verify session checks 226
verify thin and data device states 536
verify thin pool status 352
view access control 75
view all thin device pools 357
view application registrations 499
view array metricst 235
view audit log 460, 465, 466
view change tracker reports 579
view contents of vdmdb 437
view device service states 527
view external spindle information 554
View group details 403, 405
view hba alias name 406
view marked objects for change monitoring 578
view thin device allocations 358
view thin device details 356
view thin pools 356
view vcmdb HBA assignments 438
view vcmdb HBA records 437
view vdmdb 437
view vdmdb director records 437
virtual disk mapping 40
virtual LUN migration
to configured space 475
to unconfigured space 477, 478
virtual provisioning overview 332, 333
VLOGIX permission 63
vp compression 353
vp compression rules 354

W

workload distribution 458

X

XML 639
XML mode 639
XML output 639
XML output overview 638