EMC® Solutions Enabler SRDF Family

Version 8.0.3

CLI User Guide

VMAX® Family

REV 01

EMC²

Copyright © 2003 - 2015 EMC Corporation. All rights reserved. Published in the USA.

Published June 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.

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

CONTENTS

Preface

Revision history 20

Chapter 1 Introduction to the SRDF CLI

Introduction 24
SRDF documentation 25
What's new in Solutions Enabler 8.0.2 25
8.0.2 Restrictions 27
SRDF backward compatibility to Enginuity 5876 27
SYMCLI for SRDF 29
SYMCLI SRDF commands 30
symrdf command options 32
Commands to display, query and verify SRDF configurations 34
symrdf list command options 38
ping command 39
Using the verify command 39
Understanding SRDF pair states and links 42
About SRDF pair states 43
Before you begin 45
Array access rights 46
Device external locks 46
SRDF operations and copy sessions 46
Migrating R1 to a larger R2 device 46
Restricting synchronization 47
Enabling SRDF software and hardware compression 47
SRDF/A and the consistency exempt option 48
Enabling mixed-mode workloads on an SRDF director 48
FAST VP SRDF coordination 49

Chapter 2 Basic SRDF Control Operations

Basic SRDF control operations summary 52
SRDF basic control operations 55
Set mode 55
Establish (full) 58
Establish (incremental) 60
Failback 62
Failover 64
Invalidate R1 tracks 65
Invalidate R2 tracks 66
Make R1 ready 67
Make R1 not ready 67
Make R2 ready 68
Make R2 not ready 68
Merge track tables 69
Move one-half of an SRDF pair 69
Move both sides of SRDF device pairs 70
Read/write disable target device 71
Refresh R1 71

Refresh R2 72
Restore (full) 72
Restore (incremental) 74
Resume I/O on links 75
Split 76
Suspend I/O on links 78
Swap one-half of an SRDF pair 79
Swap SRDF pairs 80
Update R1 mirror 80
Write disable R1 82
Write disable R2 83
Write enable R1 83
Write enable R2 84

Chapter 3 Dynamic Operations

Overview 86
SRDF group attributes 87
Managing SRDF groups 88
Adding SRDF groups 88
Setting SRDF group attributes 91
Adding/removing supporting directors for an SRDF group 92
Removing SRDF groups 93
Device pairing operations 94
Creating a device file 94
Creating SRDF device pairs 95
Creating dynamic concurrent pairs 104
Deleting dynamic SRDF device pairs 105
Grouping, moving and swapping dynamic devices 108
Grouping dynamic pairs with a device file 108
Moving dynamic SRDF device pairs 108
Swapping SRDF devices 110
Dynamic failover operations 113

Chapter 4 SRDF/Asynchronous Operations

Overview 118
SRDF/A benefits 121
SRDF /A features 121
Restrictions 122
Using BCVs to protect the R2 side 123
Immediate option 123
SRDF/Asynchronous operations 124
Transitioning to asynchronous mode 125
Setting SRDF/A group cycle time, priority, and transmit idle 126
Checking for R1 invalid tracks 127
Enabling/disabling consistency for SRDF/A devices 128
Adding/removing devices using consistency exempt 129
Displaying checkpoint complete status 132
Managing Delta Set Extension 133
Managing the DSE SRP capacity (HYPERMAX OS) 134
Managing DSE pools (Enginuity 5773 - 5876) 135
Managing transmit idle 141
Managing SRDF/A write pacing 141
Displaying SRDF/A 148

Chapter 5 Consistency Group Operations

Overview 152
Consistency protection using the SRDF daemon 152
Redundant consistency protection 153
SRDF consistency group operations 155
Creating an SRDF consistency group 155
Creating composite groups from various sources 156
Enabling and disabling SRDF consistency protection 158
Restrictions 158
Enabling consistency: composite group vs. SRDF group name 159
Enabling SRDF consistency protection for concurrent SRDF devices 161
Checking if device pairs are enabled for consistency protection 163
Blocking symcg enable on R2 side 164
Deleting an SRDF consistency group 164
Suspending SRDF consistency protection 165
Using the msc_cleanup command 166
Modifying consistency groups 168
Restrictions 168
Preparing the staging area 169
Restrictions - add operations 172
Restrictions - remove operations 173
Adding devices to an RDF1 consistency group 174
Adding devices to a concurrent RDF1 consistency group 174
Adding devices to a cascaded RDF1 consistency group 176
Removing devices from an RDF1 consistency group 177
Removing devices from a concurrent RDF1 consistency group 178
Removing devices from a cascaded RDF1 consistency group 178
Recovering from a failed dynamic modify operation 178
Consistency groups with a parallel database 180
Consistency groups with BCV access at the target site 181

Chapter 6 Concurrent Operations

Overview 184
Restrictions 184
Other documentation 186
Concurrent SRDF operations 186
Creating/establishing concurrent SRDF devices 186
Splitting concurrent SRDF devices 187
Restoring concurrent devices 187
Viewing concurrent SRDF devices 189

Chapter 7 Cascaded Operations

Overview 192
Restrictions (cascaded SRDF) 194
Setting up cascaded SRDF 194
Setting up a cascaded SRDF relationship 194
Applicable pair states for cascaded SRDF operations 196
RDF21 SRDF groups 197
Managing R21 devices 197
Cascaded SRDF with EDP 200
SRDF/EDP rules 200
Setting up cascaded SRDF with EDP 201
Rules/restrictions for diskless devices in cascaded SRDF 203

Creating diskless devices 204
Adding a diskless SRDF mirror 205
Restarting in diskless configuration 206
Sample session: planned failover 206
Displaying/querying cascaded SRDF 208
Listing cascaded SRDF devices 208
Viewing diskless devices 209
Querying hop 2 information 211

Chapter 8 SRDF/Star Operations

Overview 216
Concurrent SRDF/Star 216
Cascaded SRDF/Star 217
SRDF/Star concurrent R22 218
SRDF/Star features 219
Restrictions 220
Understanding SRDF/Star states and operations 221
SRDF/Star state 221
Target site states 221
SRDF/Star site transitions 222
Operation categories 224
Required states for operations: Concurrent SRDF/Star 225
Required states for operations: Cascaded SRDF/Star 228
SRDF/Star operations - summary 232
symstar command options 233
Command failure while in Connected state 235
Restrictions for cascaded mode 235
Configure and bring up SRDF/Star 236
Displaying your symstar configuration 247
Basic SRDF/Star operations 251
Isolating the SRDF/Star sites 251
Unprotecting the target sites 252
Halting the target sites 252
Cleanup metadata 252
SRDF/Star consistency group operations 253
Before you begin: SRDF daemon interaction 253
Restrictions 254
Preparing the staging area 255
Adding devices to a concurrent SRDF/Star consistency group 256
Adding devices to a cascaded SRDF/Star consistency group 259
Removing devices from consistency groups 261
Recovering from a failed consistency group modification 262
Recovery and switch operations: concurrent SRDF/Star 264
Recovery operations: concurrent SRDF/Star 264
Workload switching: concurrent SRDF/Star 268
Recovery and switch operations: cascaded SRDF/Star 282
Recovery operations: cascaded SRDF/Star 282
Workload switching: cascaded SRDF/Star 286
Reconfiguration operations 296
Reconfiguring mode: cascaded to concurrent 297
Reconfiguring cascaded paths 299
Reconfiguring mode: concurrent to cascaded 301
Reconfiguring mode without halting the workload site 303
SRDF/Star configuration with R22 devices 304

Before you begin 304
Transitioning SRDF/Star to use R22 devices 305

Chapter 9 Device Migration Operations

Overview 308
Requirements 308
R1 device migration 309
R2 device migration 312
R1 and R2 migration procedures 315
Before you begin 315
Restrictions and limitations 316
Sample procedure: migrating R1 devices 317
Sample procedure: migrating R2 devices 325
SRDF pair states for migration 327
Pair states for migrate -setup 328
Pair states for migrate -replace for first leg of concurrent SRDF 329
Pair states for migrate -replace for second leg of concurrent SRDF 331

Chapter 10 SRDF/Automated Replication

Overview 336
Restrictions 336
SRDF/Automated Replication operations 337
Configuring single-hop sessions 337
Configuring multi-hop sessions 341
Concurrent BCVs with SRDF/AR 343
Setting replication cycle parameters 344
Logging symreplicate steps 346
Clustered SRDF/AR 346
Writing log files to a specified SFS 346
Restarting from another host 347
Displaying log files written to the SFS 348
Deleting a log file written to SFS 348
Setting symreplicate parameters in the options file 348
Option file format 349
Setting replication retry and sleep times 349
Setting the symreplicate control parameters 350
Managing locked devices 354
Recovering locks 354
Releasing locks 354
Acquiring persistent locks 355

Chapter 11 TimeFinder and SRDF operations

TimeFinder consistent splits in SRDF configurations 358
Enginuity Consistency Assist 358
Multi-hop operations 359
Before you begin: system-wide device groups 359
Controlling basic operations in a multi-hop configuration 361
System-wide splits 362
TimeFinder SnapVX and SRDF 363
TimeFinder SnapVX and Cascaded SRDF 363
TimeFinder SnapVX and Concurrent SRDF 364

Chapter 12 Automated Recovery Operations

Overview 368
Requirements and strategies 368
symrecover command 369
Starting SRDF Automated Recovery 370
Stopping SRDF Automated Recovery 371
symrecover options file parameters 372

Appendix A SRDF operations and pair states

SRDF operations and applicable pair states 378
Control operations for R1 -> R2 pair states 378
Cascaded SRDF control operations and applicable pair states 383
Cascaded SRDF: R1 -> R21 control operations allowed for R21-> R2 pair states 383
Cascaded SRDF: R21 -> R2 control operations allowed for R1 -> R21 pair states 386
Cascaded SRDF set operations and applicable pair states 389
Cascaded SRDF: R1 -> R21 set operations allowed for R21 -> R2 pair states . 389
Cascaded SRDF: R21 -> R2 set operations allowed for R1 -> R21 pair states . 389
Concurrent SRDF control operations and applicable pair states 390
Concurrent SRDF: control operations 1st leg R1-> 2nd leg R1 pair states 390
Concurrent SRDF: control operations 1st leg R2 -> 2nd leg R2 pair states 393
Consistency group control operations and applicable pair states 396

Appendix B SRDF operations and TimeFinder sessions

SRDF operations during TimeFinder/Snap/VP Snap sessions 398
SRDF operations when R1 is source of TimeFinder Snap/VP Snap 398
SRDF operations when R1 is target of TimeFinder Snap/VP Snap 400
SRDF operations when R2 is source of TimeFinder Snap/VP Snap 402
SRDF operations when R2 is target of TimeFinder Snap/VP Snap 405
SRDF operations for TimeFinder/Clone sessions 408
SRDF operations when R1 is source of TimeFinder Clone 408
SRDF operations when R1 is target of TimeFinder Clone 410
SRDF operations when R2 is source of TimeFinder Clone 413
SRDF operations when R2 is target of TimeFinder Clone 416
SRDF operations for Extent-level TimeFinder/Clone sessions 418
SRDF operations when R1 is source of Extent-level Clone 418
SRDF operations when R1 is target of Extent-level Clone 420
SRDF operations when R2 is source of Extent-level Clone 423
SRDF operations when R2 is target of Extent-level Clone 426
SRDF set operations for TimeFinder/Snap sessions 428
SRDF set operations when R1 is source of TimeFinder/Snap 428
SRDF set operations when R1 is target of TimeFinder/Snap 428
SRDF set operations when R2 is source of TimeFinder/Snap 429
SRDF set operations when R2 is target of TimeFinder/Snap 429
SRDF set operations for TimeFinder/Clone sessions 431
SRDF set operations when R1 is source of TimeFinder/Clone 431
SRDF set operations when R1 is target of TimeFinder/Clone 431
SRDF set operations when R2 is source of TimeFinder/Clone 432

SRDF set operations when R2 is target of TimeFinder/Clone 432
SRDF set operations for Extent-level TimeFinder/Clone sessions..... 434
SRDF set operations when R1 is source of Extent-level Clone..... 434
SRDF set operations when R1 is target of Extent-level Clone 434
SRDF set operations when R2 is source of Extent-level Clone..... 435
SRDF set operations when R2 is target of Extent-level Clone 435
SRDF operations for TimeFinder Snap VX..... 436
Determine SnapVX states-SRDF operations interaction rules 436
SRDF operations when R1 is source of Snapvx..... 437
SRDF operations when R1 is target for Snapvx..... 439
SRDF operations when R2 is source of Snapvx..... 440
SRDF operations when R2 is target for Snapvx..... 443
SRDF set operations for TimeFinder Snapvx sessions..... 444
SRDF set operations when R1 is source of TimeFinder Snapvx 445
SRDF set operations when R1 is target of TimeFinder Snapvx..... 445
SRDF set operations when R2 is source of TimeFinder Snapvx 446
SRDF set operations when R2 is target of TimeFinder Snapvx..... 446

Appendix C SRDF operations and rcopy States

Rcopy Session on the R1 side..... 448
R1 is part of an Rcopy PUSH 448
R1 is part of an Rcopy PULL 450
Rcopy Session on the R2 side..... 452
R2 is part of an Rcopy PUSH 453
R2 is part of an Rcopy PULL 455

FIGURES

	Title	Page
1	Typical 2-site SRDF configuration in open systems host environment	24
2	SRDF device and link states	42
3	Full establish	59
4	Incremental establish	61
5	Failback of an SRDF device	63
6	Failover of an SRDF device	65
7	Restore (full) an SRDF device	73
8	Incremental restore an SRDF device	75
9	Split an SRDF pair	77
10	Update SRDF device track tables	81
11	SRDF/A legacy mode	119
12	SRDF/A multi-cycle mode	120
13	Running redundant hosts to ensure consistency protection	154
14	Staging area for adding devices to the R1CG consistency group	171
15	R1CG consistency group after a dynamic modify add operation	171
16	Preparing the staging area for removing devices from the MyR1 CG	172
17	MyR1 CG after a dynamic modify remove operation	172
18	Adding a device to independently-enabled SRDF groups of a concurrent CG	175
19	Adding devices to independently-enabled SRDF groups of a cascaded CG	177
20	Using an SRDF consistency group with a parallel database configuration	180
21	Using an SRDF consistency group with BCVs at the target site	181
22	Concurrent SRDF	184
23	Concurrent SRDF/S to both R2 devices	185
24	Concurrent SRDF/A to both R2 devices	185
25	Restoring the R1 a concurrent configuration	188
26	Restoring the source device and mirror in a concurrent SRDF configuration	189
27	Cascaded SRDF configuration	192
28	Configuring the first hop	196
29	Configuring the second hop	196
30	Determining SRDF pair state in cascaded configurations	198
31	Location of hop-2 devices	199
32	Cascaded SRDF with EDP	200
33	Set up first hop in cascaded SRDF with EDP	202
34	Set up second hop in cascaded SRDF with EDP	202
35	Adding a diskless SRDF mirror	205
36	Cascaded configuration before planned failover	206
37	Planned failover - after first swap	207
38	Planned failover - after second swap	208
39	Concurrent SRDF/Star configuration	217
40	Cascaded SRDF/Star configuration	218
41	Typical concurrent SRDF/Star with R22 devices	219
42	Typical cascaded SRDF/Star with R22 devices	219
43	Site configuration transitions without concurrent devices	223
44	Site configuration transitions with concurrent devices	223
45	Concurrent SRDF/Star: Normal operations	225
46	Concurrent SRDF/Star: Transient fault operations	226
47	Concurrent SRDF/Star: Unplanned switch operations	227
48	Concurrent SRDF/Star: Planned switch operations	228
49	Cascaded SRDF/Star: Normal operations	229
50	Cascaded SRDF/Star: Transient fault operations (asynchronous loss)	230

51 Cascaded SRDF/Star: Transient fault operations (synchronous loss) 230
52 Cascaded SRDF/Star: Unplanned switch operations 231
53 Concurrent SRDF/Star setup using the StarGrp composite group 238
54 Cascaded SRDF/Star setup using the StarGrp composite group 241
55 Adding a device to a concurrent SRDF/Star CG 258
56 ConStarCG after a dynamic add operation 259
57 Adding devices to a cascaded SRDF/Star CG 260
58 CasStarCG after a dynamic add operation 260
59 Transient failure: concurrent SRDF/Star 265
60 Transient fault recovery: before reconfiguration 267
61 Transient fault recovery: after reconfiguration 268
62 Concurrent SRDF/Star: halted 270
63 Concurrent SRDF/Star: switched 270
64 Concurrent SRDF/Star: connected 271
65 Concurrent SRDF/Star: protected 272
66 Loss of workload site: concurrent SRDF/Star 273
67 Workload switched to synchronous site: concurrent SRDF/Star 275
68 Concurrent SRDF/Star: new workload site connected to asynch site 275
69 Concurrent SRDF/Star: protected to asynchronous site 276
70 Concurrent SRDF/Star: protect to all sites 277
71 Workload switched to asynchronous site: concurrent SRDF/Star 279
72 Concurrent SRDF/Star: protected to asynchronous site 279
73 Concurrent SRDF/Star: one asynchronous site not protected 281
74 transient failure: cascaded SRDF/Star 283
75 Cascaded SRDF/Star with transient fault 284
76 Cascaded SRDF/Star: asynchronous site not protected 285
77 SRDF/Star: after reconfiguration to concurrent 286
78 Cascaded SRDF/Star: halted 288
79 Cascaded SRDF/Star: switched workload site 288
80 Loss of workload site: cascaded SRDF/Star 289
81 Workload switched to synchronous target site: cascaded SRDF/Star 290
82 After workload switch to synchronous site: cascaded SRDF/Star 291
83 Cascaded SRDF/Star after workload switch: protected 292
84 After reconfiguration to concurrent mode 292
85 Protected after reconfiguration from cascaded to concurrent mode 293
86 Loss of workload site: cascaded SRDF/Star 294
87 Cascaded SRDF: after switch to asynchronous site, connect, and protect 295
88 Cascaded SRDF: after switch to asynchronous site 296
89 Halted cascaded SRDF/Star 297
90 After reconfiguration to concurrent 298
91 Halted cascaded SRDF/Star 298
92 After reconfiguration to concurrent 299
93 Halted cascaded SRDF/Star 300
94 After cascaded path reconfiguration 300
95 Halted concurrent SRDF/Star 301
96 After reconfiguration to cascaded 302
97 Halted concurrent SRDF/Star 302
98 After reconfiguration to cascaded 303
99 R1 migration: configuration setup 309
100 R1 migration: establishing a concurrent relationship 310
101 R1 migration: replacing the source device 311
102 Migrating R2 devices 312
103 R2 migration: configuration setup 313
104 R2 migration: establishing a concurrent relationship 314
105 R2 migration: replacing the target device 315

106 R1 migration example: Initial configuration 317
107 Concurrent SRDF relationship 320
108 Migrated R1 devices 323
109 R2 migration example: Initial configuration 325
110 Concurrent SRDF relationship 326
111 Migrated R2 devices 327
112 R1 migration: applicable R1/R2 pair states for migrate -setup 328
113 R2 migration: applicable R1/R2 pair states for migrate -setup 329
114 R1 migration: R11/R2 applicable pair states for migrate -replace (first leg) 330
115 R2 migration:R11/R2 applicable pair states for migrate -replace (first leg) 331
116 R1 migration: applicable R11/R2 pair states for migrate -replace (second leg) 332
117 R2 migration: applicable R11/R2 pair states for migrate -replace (second leg) 333
118 Automated data copy path in single-hop SRDF systems 337
119 Automated data copy path in multi-hop SRDF 341
120 Concurrent BCV in a multi-hop configuration 344
121 ECA consistent split 359
122 Commands used to perform splits in a complex configuration 360
123 Basic operations in multi-hop SRDF configurations 361
124 SnapVX and Cascaded SRDF 364
125 SnapVX and Concurrent SRDF 365
126 SRDF Recovery environment 368

TABLES

	Title	Page
1	Document revision history	20
2	SRDF documentation	25
3	SYMCLI SRDF commands	30
4	symrdf command options	32
5	Commands to display and verify SRDF, devices, and groups	34
6	Options for symrdf list command	38
7	SRDF pair states	43
8	Possible SRDF device and link state combinations	45
9	Access rights required by an array	46
10	SRDF control operations summary	52
11	Device type combinations for creating SRDF pairs	96
12	SRDF device states before swap operation	111
13	SRDF/A control operations	124
14	Consistency modes for concurrent mirrors	163
15	Allowable device types for adding devices to an RDF1 CG	174
16	Allowable device types for adding devices to a concurrent RDF1 CG	174
17	Supported consistency modes for concurrent SRDF groups	174
18	Allowable device types for adding devices to a cascaded RDF1 CG	176
19	Supported consistency modes for cascaded hops	176
20	Allowable device types for removing devices from an RDF1 CG	177
21	Allowable device types for removing devices from a concurrent RDF1 CG	178
22	Allowable device types for removing devices from a cascaded RDF1 CG	178
23	SRDF modes for cascaded configurations (no EDP)	193
24	SRDF modes for cascaded configurations with EDP	193
25	SRDF modes allowed for SRDF/EDP	201
26	SRDF/Star states	221
27	SRDF/Star target site states	221
28	SRDF/Star operation categories	224
29	SRDF/Star control operations	232
30	symstar command options	233
31	Allowable SRDF/Star states for adding device pairs to a concurrent CG	257
32	Allowable states for adding device pairs to an SRDF/Star cascaded CG	259
33	Pair states of the SRDF devices after symstar modifycg -add completion	261
34	Allowable states for removing device pairs from an SRDF/Star concurrent CG	262
35	Allowable SRDF/Star states for removing device pairs from a cascaded CG	262
36	Possible pair states of the SRDF devices after a recovery	264
37	SRDF migrate -setup control operation and applicable pair states	327
38	SRDF migrate -replace control operation and applicable pair states	329
39	SRDF migrate -replace control operation and applicable pair states	331
40	Initial setups for cycle timing parameters	345
41	Basic operations in a multi-hop configuration	361
42	symrecover options file parameters	372
43	SRDF control operations and applicable pair states	378
44	R1->R21 cascaded SRDF control operations and applicable pair states	383
45	R21->R2 cascaded SRDF control operations and applicable pair states	386
46	R1->R21 cascaded SRDF set operations and applicable pair states	389
47	R21->R2 Cascaded RDF Set Operations and Applicable Pair States	390
48	SRDF control operations and applicable states for concurrent R1 pairs	391
49	SRDF control operations and applicable states for concurrent R2 pairs	394
50	SRDF control operations and applicable pair states for devices in an SRDF/CG	396

51 Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap . 398
52 Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap... 400
53 Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap . 402
54 Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap... 405
55 Allowable SRDF operations when R1 is the source of a TimeFinder/Clone 408
56 Allowable SRDF operations when R1 is the target of a TimeFinder/Clone 410
57 Allowable SRDF operations when R2 is the source of a TimeFinder/Clone 413
58 Allowable SRDF operations when R2 is the target of a TimeFinder/Clone 416
59 Allowable SRDF operations when R1 is the source of an Extent-level Clone..... 418
60 Allowable SRDF operations when the R1 is the target of an Extent-level Clone 420
61 Allowable SRDF operations when R2 is the source of an Extent-level Clone..... 423
62 Allowable SRDF operations when the R2 is the target of an Extent-level Clone 426
63 Allowable SRDF set operations when R1 is the source of a TimeFinder/Snap 428
64 Allowable SRDF set operations when R1 is the target of a TimeFinder/Snap..... 428
65 Allowable SRDF set operations when R2 is the source of a TimeFinder/Snap 429
66 Allowable SRDF set operations when R2 is the target of a TimeFinder/Snap..... 429
67 Allowable SRDF set operations when R1 is the source of a TimeFinder/Clone 431
68 Allowable SRDF set operations when R1 is the target of a TimeFinder/Clone..... 431
69 Allowable SRDF set operations when R2 is the source of a TimeFinder/Clone 432
70 Allowable SRDF set operations when R2 is the target of a TimeFinder/Clone..... 432
71 Allowable SRDF set operations when R1 is the source of an Extent-level Clone 434
72 Allowable SRDF set operations when R1 is the target of an Extent-level Clone..... 434
73 Allowable SRDF set operations when the R2 is the source of an Extent-level Clone 435
74 Allowable SRDF set operations when R2 is the target of an Extent-level Clone..... 435
75 SnapVX State Determination 436
76 Allowable SRDF operations when R1 is the source of a TimeFinder Snapvx 437
77 Allowable SRDF operations when R1 is the target of a TimeFinder Snapvx..... 439
78 Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx 441
79 Allowable SRDF operations when R2 is the target of a TimeFinder Snapvx..... 443
80 Allowable SRDF set operations when R1 is the source of a TimeFinder Snapvx..... 445
81 Allowable SRDF set operations when R1 is the target of a TimeFinder Snapvx 445
82 Allowable SRDF set operations when R2 is the source of a TimeFinder Snapvx..... 446
83 Allowable SRDF set operations when R2 is the target of a TimeFinder Snapvx 446
84 Allowable SRDF operations with Rcopy PUSH session on the R1 448
85 Allowable SRDF operations with Rcopy PULL session on the R1 450
86 Allowable SRDF operations with Rcopy PUSH session on the R2 453
87 Allowable SRDF operations with Rcopy PULL session on the R2 455

PREFACE

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

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

Note: This document was accurate at publication time. New versions of this document might be released on 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 Solutions Enabler SYMCLI to manage SRDF®.

Audience

This document is for advanced command-line users and script programmers to manage various types of control operations on arrays and devices using Solutions Enabler's SYMCLI commands.

Related documentation

The following documents provide additional information about Solutions Enabler:

- *EMC Solutions Enabler V8.0.2 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 VMAX3 Family Product Guide* - Describes the VMAX3 platform and software products available.
- *EMC Solutions Enabler Array Management CLI User Guide* - Describes how to configure array control, management, and migration operations 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 provide additional information:

- *EMC VMAX V3 Family Documentation Set* - Contains documentation related to the VMAX 100K, 200K, and 400K arrays.
- *EMC VMAX Family (10K, 20K, 40K) Documentation Set* - Contains documentation related to the VMAX 10K, 20K, and 40K arrays.
- *EMC VMAX3 Family with HYPERMAX OS Release Notes* - Detail new features and any known limitations.
- *EMC VMAX Family Viewer for Desktop and iPad®* - Illustrates system hardware, incrementally scalable system configurations, and available host connectivity offered for VMAX arrays.
- E-Lab™ Interoperability Navigator (ELN) - Provides a web-based interoperability and solution search portal. You can find the ELN at https://elabnavigator.EMC.com.
- SolVe Desktop - — Provides links to documentation, procedures for common tasks, and connectivity information for 2-site and 3-site SRDF configurations. To download the SolVe Desktop tool, go to EMC Online Support at https://support.EMC.com and search for SolVe Desktop. Download the SolVe Desktop and load the VMAX Family and DMX procedure generator.

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

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file arrays, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, array calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables

‹ ›	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

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

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

Product information

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

https://support.EMC.com

Technical support

EMC offers a variety of support options.

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

https://support.EMC.com/products

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

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

eLicensing support

To activate your entitlements and obtain your license files, visit the Service Center on http://support.EMC.com, as directed on your License Authorization Code (LAC) letter e-mailed 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, contact the EMC Customer Support Center.

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

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

Your comments

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

VMAXContentFeedback@emc.com

Revision history

Table 1 lists the revision history of this document.

Table 1 Document revision history

Solutions Enabler version	New/revised content
8.0.3	New restriction for the "update" operation when the R2 device is on an array running OS 5977. New restrictions for "createpair -establish", "createpair -restore", "createpair -invalidate R1", "createpair -invalidate R2", and "createpair -format" operations when R1 device is part of an Rcopy PUSH operation.
8.0.2	New content: • "Cascaded Operations" on page 191 • "SRDF/Star Operations" on page 215 • "SRDF operations for TimeFinder Snap VX" on page 436 • "SRDF set operations for TimeFinder Snapvx sessions" on page 444 Changed/updated content: • Storage groups (SGs) are a collection of devices stored on the array that are used by an application, a server, or a collection of servers. "Creating pairs with -sg option" on page 98
8.0.1	New content • Track size increased to 128K: Geometry Compatibility Mode supports full SRDF functionality for devices of different track sizes. "Geometry Compatible Mode" on page 28 Changed/updated content: • Multi-core, multi-ports per director: Specify both the director AND the ports for the SRDF emulation to use on each side. "symrdf addgrp command syntax" and "symrdf modifygrp command syntax" Removed content (deprecated commands and options): • symrdf monitor command, including all options and arguments. symcfg monitor command provides identical functionality. • -port option removed from the symcfg -ra all list command. • -vxfs and -rdb options removed from the symreplicate start command. • SYMAPI_SYNC_DIRECTION option removed from the options file.

Table 1 Document revision history

Solutions Enabler version	New/revised content
7.6	New content: SRDF devices can be tagged for use with RecoverPoint. A new flag **-rp** has been added to the `symrdf` command. For more information, see white paper *VMAX RecoverPoint Splitter and Enhancements in Enginuity 5876* Revised content: SRDF pair state tables were updated: "SRDF operations and pair states" on page 377 Removed content: The following chapters were moved from this guide to the SolVe Desktop VMAX Family and DMX procedure generator: • Performing SRDF Control Operations • Querying and verifying with SRDF Commands • Implementing Consistency Protection • Performing SRDF/Automated Replication Operations The steps to download SolVe and the generator are described in "Related documentation" on page 17

CHAPTER 1
Introduction to the SRDF CLI

This chapter describes the following topics:

- Introduction 24
- SRDF documentation 25
- What's new in Solutions Enabler 8.0.2 25
- SYMCLI for SRDF 29
  - "SYMCLI SRDF commands" on page 30
  - "symrdf command options" on page 32
  - "Commands to display, query and verify SRDF configurations" on page 34
  - "symrdf list command options" on page 38
  - "ping command" on page 39
  - "Using the verify command" on page 39
- Understanding SRDF pair states and links 42
- Before you begin 45

Introduction

SRDF® is a family of disaster recovery, parallel processing, and data migration solutions.

SRDF configurations require at least two arrays; a *primary* array and at least one *secondary* array. The arrays can be located in the same room, in different buildings within the same campus, or hundreds, even thousands of kilometers apart.

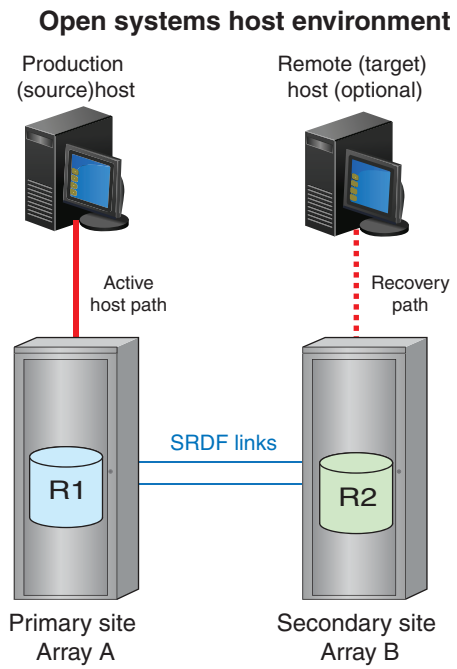


Figure 1 Typical 2-site SRDF configuration in open systems host environment

In the **open systems host** environment:

- A control host at the production site is connected to the primary array.
- Primary and secondary sites are connected other over SRDF links.
- The production host writes I/O to devices (R1) at the primary site.
- SRDF mirrors the production I/O to the devices (R2) at the secondary site(s).

HYPERMAX OS

VMAX 100K/200K/400K arrays (referred to as VMAX3™ arrays) running HYPERMAX OS can use SRDF to replicate to:

- VMAX3 arrays running HYPERMAX OS.
- VMAX 10K/20K/40K arrays running Enginuity™ version 5876 with applicable ePack.

Enginuity versions 5773 - 5876

Refer to the *SRDF Two-site Interfamily Connectivity* tool for information about SRDF features supported between arrays running Enginuity versions 5773 through 5876.

SRDF documentation

Table 2 SRDF documentation

For information on	See
Technical concepts and operations of the SRDF product family. Topics include: • SRDF Solutions • SRDF interfamily connectivity • SRDF concepts and terminology • SRDF/DM, SRDF/AR, SRDF/Concurrent • SRDF integration with other products	*EMC VMAX3 Family Product Guide*
Configure and manage arrays using the SYMCLI.	*EMC Solutions Enabler Array Management CLI User Guide*
Determine which SRDF replication features are supported between two or three arrays running Enginuity 5773-5876, or HYPERMAX OS.	Download the SolVe Desktop and load the VMAX Family and DMX procedure generator. Select *SRDF 2-site Interfamily connectivity* or *SRDF 3-site Interfamily connectivity.*
Securing your configuration	*EMC VMAX Family Security Configuration Guide*
Host connectivity	EMC Host Connectivity Guides for your operating system.
Managing legacy versions of SRDF using SYMCLI	Download the SolVe Desktop and load the VMAX Family and DMX procedure generator. Select VMAX 10K, 20K, 40K, DMX -> Customer procedures -> *Managing SRDF using SYMCLI.*

What's new in Solutions Enabler 8.0.2

This section lists changes introduced by Solutions Enabler 8.0.2/HYPERMAX OS.

Cascaded SRDF

Cascaded SRDF configurations can include one or more arrays running Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR.

An array running Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR supports device types R1, R2, and R21 and can connect to:

- VMAX 3 arrays also running HYPERMAX OS.
- VMAX 10K/20K/40K arrays running Enginuity 5876 with the applicable ePack.

"Cascaded Operations" on page 191 provides more information.

Concurrent and Cascaded SRDF/Star

Concurrent and Cascaded SRDF/Star configurations can include one or more arrays running Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR.

An array running Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR supports device types R1, R2, R21, and R22 and can connect to:

- VMAX 3 arrays also running HYPERMAX OS.
- VMAX 10K/20K/40K arrays running Enginuity 5876 with the applicable ePack.

IMPORTANT

Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR includes changes to the format of the Star definition file. Older versions of Solutions Enabler cannot list/query/show/control Star CGs on arrays running Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR.

"SRDF/Star Operations" on page 215 provides more information.

Storage groups

Storage groups (SGs) are a collection of devices stored on the array that are used by an application, a server, or a collection of servers. *Solutions Enabler Array Management CLI User Guide* provides more information about storage groups.

symrdf createpair operations have been modified to support storage groups. "Creating pairs with -sg option" on page 98 provides more information.

symrdf control and set operations that support the -file *Filename* option have been modified to support the same operations on a storage group.

The following command options have been added or modified:

- - **sg** *SGName* **-** Name of storage group on the local array. Required for all -sg operations.
-
- -**hop2_rdfg** *GroupNum* - SRDF group for the second hop. Used with -sg createpair -hop2.
- -**rdfg** *GroupNum* - SRDF group associated with the SG. Required for all -sg operations.

  Note: -hop2_rdfg *GroupNum* specifies the SRDF group at hop 2 used to create the new hop2 pair.

- **-remote_sg** *SGName* - Name of the storage group on the remote array. Used only for createpair operations.

The following topics provide additional information:

- "Pairing devices using storage groups" on page 98
- "Pairing mixed devices using storage groups" on page 100
- "Pairing devices in cascaded storage groups" on page 100
- "Pairing devices in storage groups (second hop)" on page 101

Operations (other than createpair) that have a mixture of RDF and non-RDF devices, or different RDF types (R1, R2) are not allowed.

Note: Createpair operations for SGs that have a mixture of RDF and non-RDF devices, or a set of RDF devices that have a mixture of different RDF types is allowed.

If devices in the SG have 2 SRDF mirrors, two commands are required, one for each mirror.

Cascaded SGs contain one or more child SGs. For Cascaded SGs, operations work on all devices in the specified scope. If the scope is a parent SG containing multiple child SGs, operations work on all devices in all child SGs.

RDF operations that use SG as the selection type select only devices which are not considered to be gatekeepers.

8.0.2 Restrictions

- Solutions Enabler 7.6/Enginuity 5876 cannot discover arrays running Solutions Enabler 8.0.2/HYPERMAX OS.
- When one side of an SRDF link is running HYPERMAX OS, and the other side is running Enginuity 5876, the RDF pair state is Partitioned.

SRDF backward compatibility to Enginuity 5876

You can use SRDF features in Solutions Enabler 8.0.2/HYPERMAX OS to replicate to/from:

- VMAX 3 arrays also running HYPERMAX OS.
- VMAX 10K/20K/40K arrays running Enginuity 5876 with the applicable ePack.

When one array in an SRDF configuration is running HYPERMAX OS, and one or more other arrays are running Enginuity 5876, the following rules and restrictions apply:

- All SRDF groups and devices must be dynamic.
- SRDF/A sessions use legacy mode. See "SRDF/A cycle modes" on page 118.
- Directors on arrays running HYPERMAX OS support up to 16 ports and 250 SRDF groups. If a port on the array running HYPERMAX OS is connected to an array running Enginuity 5876:
  - The port supports a maximum of 64 RDF groups.
  - The director associated with the port supports a maximum of 186 RDF groups.
- SRDF device pairs with meta-devices on one side are allowed if the meta-devices are on the array running Enginuity 5876.

  Output of the **symrdf query**, **symrdf list**, and **symdev show** commands has been enhanced to display RDF mode as MIXED when a meta head device on an array running Enginuity 5876 has different RDF modes than its members.

  When you see a device in MIXED mode, you can then use the **set mode** command to choose the appropriate mode for the device pair.
- The **symcfg list -ra** command has been modified to report the remote SID when the RDF Pair State is Partitioned.
- Adaptive copy write pending is not supported in HYPERMAX OS.

- For **swap** and **failover** operations - If the R2 device is on an array running HYPERMAX OS, and the mode of the R1 is adaptive copy write pending, SRDF sets the mode to adaptive copy disk.
- For **migrate -replace R1** operations - If the R1 (after the replacement) is on an array running HYPERMAX OS, and the mode of the R1 is adaptive copy write pending mode, SRDF sets the mode of the migrated pair to adaptive copy disk.

Geometry Compatible Mode

An array running HYPERMAX OS cannot create a device that is exactly the same size as a device with an odd number of cylinders on an array running Enginuity 5876. In order to support the full suite of features:

- SRDF requires that R1 and R2 devices in a device pair be same size.
- TimeFinder requires that source and target devices are the same size.

HYPERMAX OS manages the size difference without user intervention, using a new device attribute, Geometry Compatible Mode (GCM). A device with GCM set is treated as half a cylinder smaller than its true configured size, enabling full functionality between HYPERMAX OS and Enginuity 5876 for SRDF, TimeFinder SnapVX, and TimeFinder emulations (TimeFinder/Clone, TimeFinder VP Snap, TimeFinder/Mirror), and ORS.

You can use the **set** command to manually set the GCM attribute for devices on V3 arrays. However, for most operations, Solutions Enabler sets it automatically when required. For example, Solutions Enabler automatically sets the GCM attribute when restoring from a physically larger R2.

- The **symdev set** /**unset**, **symdg set**/**unset**, **symcg set**/**unset**, and **symsg set**/**unset** commands have been enhanced with a new option **-gcm** to set and unset GCM for a device or group.
- The **symrdf createpair** command has been enhanced to transparently set/unset the GCM attribute as part of the create pair operation, as follows:
  - Set the GCM attribute for a target device that is configured ½ a cylinder larger. The source of the copy can be:
    - A device on an array running Enginuity 5876 with an odd number of cylinders and capacity that matches the GCM size of the target device.
    - A GCM device on an array running HYPERMAX OS.
  - Unset the GCM attribute for a target device that is configured the exact same size as the source of the copy. The source of the copy can be:
    - A source device on an array running Enginuity 5876 with even number of the cylinders and capacity that matches the size of the target device on the array running HYPERMAX OS
    - A source device on the array running HYPERMAX OS without the GCM attribute.

  IMPORTANT

  The GCM setting for a device cannot be changed if the target of the data device is already part of another replication session.

- The **symdev show**, **symdev list –v**, **symdg show ld**, **symdg list ld –v**, **sympd show**, and **sympd list –v** commands have been enhanced to report the GCM attribute.

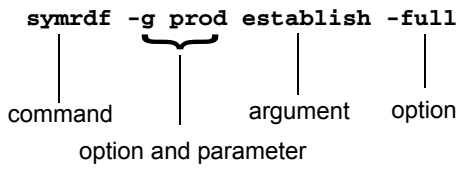
SYMCLI for SRDF

This section describes:

- "SYMCLI command syntax"
- "Getting command help"
- "Setting environmental variables"
- "Presetting names and IDs"
- Table 3 on page 30 lists the four main SRDF SYMCLI commands to establish, maintain and monitor SRDF configurations.
- Table 4 on page 32 lists options for the symrdf command.
- Table 5 on page 34 lists a variety of commands to display, query and verify your SRDF configuration.
- Table 6 on page 38 lists options for the symrdf list command

SYMCLI command syntax

The following example shows the command syntax for initiating a full establish for the SRDF pairs in the `prod` device group.



Getting command help

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

To display help for the symrdf command:

```
symrdf - h
```

On UNIX arrays, type `man` *`command`* to display the man page for the specified command.

To display the man page for the symrdf command:

```
man symsrdf
```

- **On UNIX hosts**: specify the SYMCLI man page directory (/usr/symcli/man/) in the SYMCLI_MANPATH environment variable.
- **On Windows hosts**: the default directory for man pages is C:\Program Files\EMC\symcli\man

Setting environmental variables

SYMCLI includes variables to streamline command line sessions.

To display a list of variables that can be set for your SYMCLI session:

```
symcli -env
```

To view the variables that are set:

```
symcli - def
```

To set a variable, type `setenv VARIABLE_NAME value`:

```
setenv SYMCLI_VERBOSE 1
```

To turn off a variable, type `unsetenv VARIABLE_NAME`:

```
unsetenv SYMCLI_VERBOSE
```

Presetting names and IDs

Use the SYMCLI environmental variables to preset the identity of objects, such as SID. Once the object's identity is defined, you do not need to type them in the command line.

To set the SID for all **-sid** arguments:

```
set env SYMCLI_SID 000192601365
```

To view a list of environment variables that can be set for a given SYMCLI session:

```
symcli -env
```

To view the current setting for all environment variables:

```
symcli -def
```

SYMCLI SRDF commands

Table 3 SYMCLI SRDF commands (page 1 of 2)

Command	Description	For more information
`symrdf`	Control operations on SRDF devices, including: • Establishes (mirrors) an SRDF pair by initiating a data copy from the source (R1) side to the target (R2) side. This operation can be a full or incremental establish. • Restores remote mirroring. Initiates a data copy from the target (R2) side to the source (R1) side. This operation can be a full or incremental restore. • Splits an SRDF pair, which stops mirroring for the SRDF pairs in a device group. • Fails over and back from the source (R1) side to the target (R2) side, switching data processing to the target (R2) side. • Updates the source (R1) side after a failover, while the target (R2) side may still be operational to its local host(s). • Swaps the source (R1) and target (R2) destinations between the target and the source. • Creates, deletes, or swaps dynamic SRDF device pairs. • Performs dynamic SRDF group controls to add, modify, and remove dynamic groups. • Enables link domino locally or remotely when creating dynamic groups. • Enables auto link recovery locally or remotely when creating dynamic groups. • Enables/disables consistency for SRDF/A capable devices operating in asynchronous mode that are managed by a device group or file.	See: • "Basic SRDF control operations summary" on page 52 • "SRDF basic control operations" on page 55 • `symrdf` man page.
`symreplicate`	Invokes a replicate session that generates automated recurrent, background copies of the standard data following a path across SRDF links and cascading BCVs. Start, stop, and restart a replicate session. Used for SRDF/Automated Replication.	See: • "Automated Recovery Operations" on page 367 • `symreplicate` man page.

Table 3 SYMCLI SRDF commands (page 2 of 2)

Command	Description	For more information
`symstar`	Uses concurrent SRDF/Synchronous and SRDF/Asynchronous links to replicate source data synchronously to a nearby regional site and asynchronously to a distant remote site.	See: • "SRDF/Star Operations" on page 215 • `symstar` man page.
`symrecover`	Monitor the session state during attempts to restart a group session if it enters the suspended or partitioned state.	See: • "symrecover command" on page 369 • `symrecover` man page.

symrdf command options

Note: Refer to the `symrdf` man page for more detailed descriptions of the options used with the symrdf command.

Table 4 symrdf command options (page 1 of 3)

Command option	Description
`-all`	Targets the SRDF action at all devices in the device group, which includes standard SRDF devices and any BCV SRDF devices that are locally associated with the device. When used with `list`, the `-all` option shows all SRDF mirrors of the selected devices. **Note:** The -all flag is not supported for SRDF control operations on device groups or composite groups with type ANY.
`-autostart`	Specifies whether SRDF/A DSE is automatically activated when an SRDF/A session is on (Enabled) or off (Disabled) for the SRDF group. Valid values are on (Enabled) or off (Disabled). **Note:** AutoStart for DSE is enabled by default in HYPERMAX OS.
`-bcv`	Targets the specified BCV devices associated with a device or composite group and are configured as SRDF BCV devices. By default, only the SRDF standard devices are affected by the SRDF control operations.
`-brbcv`	Targets the SRDF action at the specified remotely associated SRDF (Hop 2) BCV devices that can be paired with the remote mirrors of the local BCV devices.
`-both_sides`	Targets the SRDF control operation at both sides of an SRDF link.
`-bypass`	Causes the SRDF control operation to bypass existing exclusive locks. Use this option ONLY if no other SRDF operation is in progress at either the local and/or remote arrays.
`-c`	Counts the number of times to display or to attempt acquiring exclusive locks on the host database, the local array, and the remote arrays. If the (`-c`) option is not specified and an interval (`-i`) is specified, the program loops continuously to produce infinite redisplays, or until the SRDF control or `set` operation starts.
`-cg`	Specifies the composite group for SRDF operations.
`-cons_exempt`	For an SRDF group supporting an active SRDF/A session, allows devices to be added, removed, or suspended without affecting the state of the SRDF/A session or requiring that other devices in the session be suspended to perform the control operation. When used with `list`, this option shows devices that are consistency exempt or that are paired with devices that are consistency exempt.
`-fibre`	Uses the Fibre Channel communication protocol.
`-file` *`Filename`*	Specifies the device file for SRDF operations.
`-force`	Performs the control operations on SRDF devices that are not in the expected state for a control operation. By using this option, the control operation is attempted, regardless of the pair state of the SRDF devices, and according to the rules in Table 43 on page 378.
`-format`	Used with `createpair` to clear all tracks on the R1 and R2 sides, ensuring no data exists on either side, and makes the R1 read write to the host.
`-full`	Requests a full establish or restore operation.
`-g` *`GroupName`*	Specifies the device group for SRDF operations.
`-h`	Provides brief, online help.
`-hop2`	For cascaded configurations, specifies a group's second-hop devices.

Table 4 symrdf command options (page 2 of 3)

Command option	Description
-hop2_rdfg	Used with the createpair command that specifies a storage group. Specifies the SRDF group number at the second hop. Used only with createpair -hop2 when creating pairs using storage groups.
-hwcomp	Enables or disables hardware compression, which minimizes the amount of data to transmit over an SRDF link.
-i	Executes a command at repeat intervals to display information or to attempt to acquire an exclusive lock on the host database, the local array, and the remote arrays. The default interval is 10 seconds. The minimum interval is 5 seconds.
-immediate	Applies only to SRDF/A-backed devices. Causes `failover`, `split`, and `suspend` actions to drop the SRDF/A session immediately.
-label	Specifies a label for a dynamic SRDF group.
-noecho	Suppresses the display of progress status information.
-noprompt	Suppresses the message asking you to confirm an SRDF control operation.
-nowd	Bypasses the check to ensure the target of the operation is not writable by the host.
-offline	Obtains the data strictly from the configuration database. No connections are made to any arrays. The `symrdf` command uses information previously gathered from the array and held in the host database as opposed to interrogating the array directly. The offline option can alternatively be set by assigning the environment variable SYMCLI_OFFLINE to 1.
-rdfa_devpace	Indicates the operation affects the SRDF/A device-level write pacing feature.
-rdfa_dse	Indicates the operation affects the SRDF/A Delta Set Extension (DSE) feature.
-rdfa_pace	Indicates the operation affects both the group-level and the device-level components of the SRDF/A write pacing feature.
-rdfa_wpace	Indicates the operation affects the SRDF/A group-level write pacing feature.
-rdfa_wpace_exempt	Excludes the specified devices from SRDF/A group-level write pacing.
-rdfg	Targets a specific SRDF group number. When used **-sg createpair -hop2**, identifies the SRDF group associated with the specified storage group. **Note:** -hop2_rdfg specifies the SRDF group used to create the hop2 pair.
-rdf_mode	Used in `createpair` to set the SRDF mode of device pairs to one of the following: synchronous (sync), semi-synchronous (semi), asynchronous (async), adaptive copy disk mode (acp_disk), or adaptive copy write pending mode (acp_wp). **Note:** Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.
-refresh	Marks the source (R1) devices or the target (R2) devices to refresh from the remote mirror.
-remote	Requests a remote data copy with the `failback`, `restore`, `resume`, `createpair` and `update` actions. When the concurrent links are ready, data is also copied to the concurrent SRDF mirror. For these actions to execute, use this option or suspend the concurrent links.
-remote_rdfg	Specifies the SRDF group number for the remote array.

Table 4 symrdf command options (page 3 of 3)

Command option	Description
`-remote_sg`	Specifies the remote storage group name. Used with **createpair** to specify the storage group. Used with **createpair -hop2** to specify the storage group at the second hop.
`-remote_sid`	Specifies the remote array ID.
`-restore`	Used with `failover` to swap the R1 and R2 and restore the invalid tracks on the new R2 side (formerly R1) to the new R1 side (formerly R2). For more information, refer to "Dynamic failover restore" on page 114.
`-rp`	Used with `-establish|-restore, createpair, failback, merge, restore, resume, update`, and `refresh`, to allow the operation even when one or more devices are tagged for RecoverPoint. When used with refresh, only allowed for refresh R1.
`-rrbcv`	Targets the SRDF action at the specified remotely associated SRDF (Hop 2) BCV devices, which can be paired with the remote mirrors of the local standard devices.
`-sg`	Specifies a storage group for SRDF operations.
`-sid`	Specifies the local array ID.
`-swcomp`	Enables or disables software compression, which minimizes the amount of data to transmit over an SRDF link.
`-symforce`	Requests that the array force an operation by overriding all instances causing the array to reject an operation. The SYMAPI_ALLOW_RDF_SYMFORCE setting in the options file must be set to TRUE to use -symforce. With -symforce, a split command executes on an SRDF pair, even during a sync in progress state. **Note:** Use caution when applying this option as data can become lost or corrupted.
`-until`	Checks the number of invalid tracks that are allowed to build up from the active R2 local I/O before another update (R2 to R1) copy is retriggered. The update sequence loops until the invalid track count is less than the number specified for the `-until` value. Refer to "Continuous R1 updates" on page 82 for more information.
`-v`	Provides more detailed, verbose command output.

Commands to display, query and verify SRDF configurations

Table 5 lists SYMCLI commands to display, query, and verify your SRDF configuration.

Table 5 Commands to display and verify SRDF, devices, and groups

SYMCLI command	Description of command output
symcfg list	
`symcfg list`	Displays the connectivity (Local or Remote) of each array. Useful for verifying that only one array is connected to the host in a SRDF/Star configuration.
`symcfg list -v`	Displays a more detailed (verbose) listing, including: • Concurrent SRDF Configuration State • Dynamic SRDF Configuration State • Concurrent Dynamic SRDF Configuration • RDF Data Mobility Configuration State

Table 5 Commands to display and verify SRDF, devices, and groups

SYMCLI command	Description of command output
`symcfg list -sid SID -rdfg {all|RDF_group_number}`	Displays SRDF group-level settings for a specific group or all groups on a array, such as: • Group type • Director configuration • Group flags, including auto link recovery, link domino, SRDF/Star mode, SRDF software and hardware compression, and SRDF single round trip • SRDF flags, including consistency and SRDF status and mode

`symcfg list -RA {all|Director}`
 	
`symcfg list -RA {all|Director} -rdfg RDF_group_number`
	Display all RDF directors, or a a specified RDF director.
	Display RDF directors associated with a specified SRDF group.

`symcfg list -RA {all|Director} -p {all|Port}`	HYPERMAX OS only. Display all ports or specified port for SRDF groups configured on all or the specified director.: • Port ID • Negotiated speed (Gb/second) • Maximum speed (Gb/second) • Port status (online or offline)
symdev list	
`symdev list -r1`	Displays only the R1 side of the SRDF configuration. R1 devices not in a device group are displayed as N/Grp'd.
`symdev list -sid SID -r1 -bcv`	Displays the RDF1 BCV devices for the specified array.
`symdev list -sid SID -devs Device:Device -lock`	Display devices with a device external lock. Displays a specified range of devices that have a device external lock.
symdev show	
`symdev show Device_number -sid SID`	Displays information about the specified SRDF devices, including: • SRDF device type and its group number • Whether it is paired with a diskless or concurrent device • Whether it has a standard/thin relationship • If the R2 device is larger than its R1 • Whether SRDF/A group-level and/or device-level write pacing is currently activated and supported for the SRDF/A session • Whether it is pace-capable
symdg show	
`symdg show group_name`	Displays detailed information about device groups, including RDF groups.
symmir query	
`symmir -g RDF_group_name query`	Displays the BCV pairs in the specified device group and their state of mirroring.
symrdf list	

Table 5 Commands to display and verify SRDF, devices, and groups

SYMCLI command	Description of command output
`symrdf list`	Displays the SRDF configuration, including source devices, remote target devices, and whether a device is an R1 or R2, SRDF group, replication method, pair state, invalid tracks, and the state of each device and the SRDF links that connect them. See Table 6, "Options for symrdf list command," for a list of symrdf list command options.
symrdf query	
`symrdf -g` *`RDF_device_group_name`* `query`	Displays the state of the SRDF devices and their SRDF links in the specified device group. During normal operations, the SRDF pair is Synchronized: • The R1 devices and SRDF links are read-writable. • The R2 devices are write disabled. • The link is in synchronous replication. During failed over operations: • The R1 devices are write disabled. • The R2 devices are read/write. • The SRDF links are suspended.
`symrdf -g` *`RDF_device_group_name`* `query -all`	Displays the SRDF pair state of all devices in the specified device group, regardless of the device type.
`symrdf -g` *`RDF_group_name`* `query -bcv`	Displays the SRDF pair state of the SRDF BCV devices in the specified device group.
`symrdf -g` *`RDF_device_group_name`* `query -summary`	Displays summarized information about the state of the SRDF devices and their SRDF links in the specified device group, including: • Pair state • Number of invalid tracks on the source and target • Synchronization rate • Estimated time remaining for SRDF pair synchronization.
`symrdf -cg` *`composite_group_name`* `query`	Displays the state of the SRDF devices and their SRDF links in the specified composite group.
`symrdf -sid` *`SID`* `-rdfg` *`GrpNum`* `-sg` *`storage_group_name`* `query`	Displays the state of the SRDF devices and their SRDF links in the specified storage group.
symrdf verify (group)	
`symrdf -g` *`RDF_group_name`* `verify`	Verifies/displays the state of devices in the specified device group.
`symrdf -g` *`RDF_group_name`* `verify -failedover`	Verifies/displays whether any devices in the specified device group are in the 'Failed Over' state.
`symrdf -g` *`RDF_group_name`* `verify -synchronized`	Verifies/displays whether any devices in the specified device group are in the 'Synchronized' state.
`symrdf -g` *`RDF_group_name`* `verify -i 30 -synchronized`	Verifies/displays a message every 30 seconds as to whether any devices in the specified device group are in the 'Synchronized' state.
`symrdf -g` *`RDF_group_name`* `verify -all -i 5 -synchronized`	Verifies/displays a message every 5 seconds as to whether any devices in the specified device group are in the 'Synchronized' state until all SRDF pairs are synchronized.
`symrdf -g` *`RDF_group_name`* `verify -split`	Verifies/displays whether any devices in the specified device group are in the 'Split' state.

Table 5 Commands to display and verify SRDF, devices, and groups

SYMCLI command	Description of command output
`symrdf -g RDF_group_name verify -syncinprog`	Verifies/displays whether any devices in the specified device group are in the 'SyncInProg' state.
symrdf verify (composite group)	
`symrdf -cg composite_group_name verify`	Displays the state of devices in the specified composite group.
`symrdf -cg composite_group_name verify -consistent`	Verifies/displays whether devices in the specified composite group are in the 'Consistent' state.
`symrdf -cg composite_group_name verify -consistent -noinvalids -i 60`	Monitors and reports (one line message) the clearing of invalid tracks. Verifies/displays a one-line message every 60 minutes as to whether any devices in the specified composite group are in the 'Consistent with no invalid tracks' state until all SRDF pairs in the group are the "Consistent with no invalid tracks" state.
`symrdf verify -summary -consistent -noinvalids -cg composite_group_name -i 45`	Monitors and reports (detailed message) the clearing of invalid tracks. Verifies/displays a detailed message every 45 minutes as to whether any devices in the specified composite group are in the 'Consistent with no invalid tracks' state until all SRDF pairs in the group are the "Consistent with no invalid tracks" state.
symrdf verify (storage group)	
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify`	Verifies/displays the state of devices in the specified storage group.
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -failedover`	Verifies/displays whether any devices in the specified storage group are in the 'Failed Over' state.
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -synchronized`	Verifies/displays whether any devices in the specified storage group are in the 'Synchronized' state.
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify –i 30 -synchronized`	Verifies/displays a message every 30 seconds as to whether any devices in the specified storage group are in the 'Synchronized' state.
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -all -i 5 -synchronized`	Verifies/displays a message every 5 seconds as to whether any devices in the specified storage group are in the 'Synchronized' state until all SRDF pairs are synchronized.
`symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -split`	Verifies/displays whether any devices in the specified storage group are in the 'Split' state.
symstar list	
`symstar list`	Displays all the SRDF/Star composite groups visible to the host.
`symstar list -local`	Displays all the SRDF/Star composite groups local to your host.
symstat command options	
`-rdfg #|ALL`	Collect/display statistics for SRDF/A sessions by specified RA group number or all groups.
`-RepType rdf|rdfa`	Collect/display statistics for the specified replication type.

Table 5 Commands to display and verify SRDF, devices, and groups

SYMCLI command	Description of command output
`-type REQUESTS|CACHE|CYCLE|RDF`	Collect/display specified statistic type. REQUEST (default) -I/O requests and throughput for device(s) and director(s). CACHE - Cache activity for selected front-end or remote link director(s). CYCLE - Active SRDF/A sessions or sessions that have non-zero cache usage. RDF - SRDF/A sessions.
`-RE|-RA|ALL`	Collect/display statistics for the specified SRDF director (-RA), GigE SRDF director (-RE) or both (ALL).
`-rdflink`	Collect/display SRDF link-level statistics.
`-rdf_nw_comp`	Collect/display SRDF network compression (iSCSI statistics).
`-rdf_spdlmt`	Collect/display SRDF speed limit information (iSCSI statistics).

symrdf list command options

Table 6 lists options for the `symrdf list` command, and describes the resulting output.

Table 6 Options for symrdf list command (page 1 of 2)

symrdf list option	Description of output
`-all`	Lists all mirrors of the selected SRDF devices.
`-bcv`	Lists only BCV devices.
`-both`	Lists all SRDF devices that are RDF1 or RDF2 capable, when used with -dynamic.
`-c`	Specifies the number (count) of times to repeat the operation, displaying results appropriate to the operation at each iteration.
`-concurrent`	Lists concurrent SRDF (RDF11, RDF22, and RDF21) devices and the SRDF devices paired with a concurrent SRDF device. When used with `-R1`, lists RDF11 devices and RDF1 devices that are paired with a concurrent SRDF device. When used with `-R2`, lists RDF22 devices and RDF2 devices that are paired with a concurrent device.
`-consistency`	Displays the SRDF consistency state when listing SRDF devices. To show the consistency state in the list of all the SRDF devices in array 333: `symrdf -sid 333 -consistency list`
`-cons_exempt`	Lists devices that are consistency exempt or are paired with devices that are consistency exempt.
`-dir`	Lists the local directors (separated by commas), such as, 1a, 1b, and so on.
`-diskless_rdf`	Lists diskless SRDF devices and the devices paired with diskless SRDF devices. When used with `-R1`, lists RDF1 devices that are either diskless or that are paired with a diskless device. When used with `-R2`, lists RDF2 devices that are either diskless or are paired with a diskless device. When used with `-R21`, lists RDF21 devices that are either diskless or that are paired with a diskless device.

Table 6 Options for symrdf list command (page 2 of 2)

symrdf list option	Description of output
`-dup_pair`	Lists SRDF devices that are paired with the same SRDF type. To list all of the duplicate pair devices in array 333: `symrdf -sid 333 -dup_pair list` **Note:** Duplicate pair devices can result from an SRDF/Star failover scenario or a configuration change.
`-dynamic`	Lists devices configured as dynamic SRDF. Use the qualifiers of `-R1`, `-R2`, or `BOTH` to restrict the display to the specified device type.
`-half_pair`	Lists devices whose partner is not an SRDF device. To list all of the half pair devices in array 333: `symrdf -sid 333 -halfpair list` **Note:** Half pair devices can result from an SRDF/Star failover scenario, a `half_deletepair` operation, or a configuration change.
`-nobcv`	Lists standard SRDF devices only (excludes SRDF BCV devices).

`-R1`
`-R2`
`-R21`
	Lists devices of RDF1 types (`-R1`), RDF2 types (`-R2`), or RDF21 types (`-R21`), respectively.

`-rdfa`	Lists devices that are SRDF/A-capable.
`-rdfa_not_pace_capable`	Lists devices participating in the SRDF/A session that are not pace-capable
`-rdfa_wpace_exempt`	Lists devices that are exempt from group-level write pacing
`-rdfg`	Lists all devices within a specified SRDF group.
`-resv`	Lists SRDF devices with SCSI reservations. To list all the SRDF devices in array 333 that have SCSI reservations: `symrdf -sid 333 -resv list`
`-star_mode`	Lists device that are SRDF/Star protected. For more information, refer to the *EMC VMAX3 Family Product Guide.*

ping command

Use the `symrdf -rdf ping` command to determine if a array using SRDF links is up and running.

To ping SID 123:

```
symrdf -rdf -sid 123 ping
```

The return codes tell you whether some or all of the arrays were successfully pinged.

For more information on return codes, refer to the *EMC Solutions Enabler CLI Command Reference*.

Using the verify command

Use the `symrdf verify` command to verify the SRDF mode and pair states of device groups, composite groups, and device files.

Use the `symrdf verify -enabled` command to verify that device pairs are enabled for consistency protection.

Verify SRDF mode

When verifying two or more SRDF modes using one command, Solutions Enabler logically ORs each mode to determine the result.

Example In the following example, a device group named STAGING contains devices in synchronous (-sync), and adaptive copy disk (-acp_disk) modes, but no devices in asynchronous (-async) mode.

- If the verify command specifies only asynchronous mode:

  `symrdf -g STAGING -rdfg 129 verify -async`

  None of the device pairs in STAGING are in asynchronous mode, and the following message is displayed:

  ```
  None of the devices in the group 'STAGING' are in 'Asynchronous'
  mode.
  ```

- If the verify command specifies asynchronous, synchronous mode, OR adaptive copy disk mode:

  `symrdf -g STAGING -rdfg 129 verify -async -sync -acp_disk`

  All device pairs in STAGING are using synchronous OR adaptive copy disk mode. The following message is displayed, even though NO devices are in asynchronous mode:

  ```
  All devices in the group 'STAGING' are in 'Asynchronous, Synchronous,
  Adaptive Copy Disk' modes.
  ```

Verify SRDF pair states

When verifying two or more SRDF pair states using one command, Solutions Enabler logically ORs each pair state to determine the result.

Example In the following example, a device group named STAGING contains devices in -split, -suspended, and -synchronized states, but no devices in -consistent state.

- If the verify command specifies only Consistent state:

  `symrdf -g STAGING -rdfg 129 verify -consistent`

  None of the device pairs in STAGING are in the Consistent state, and the following message is displayed:

  ```
  None of the devices in the group 'STAGING' are in 'Consistent' state.
  ```

- If the verify command specifies Consistent OR Split state:

  `symrdf -g STAGING -rdfg 129 verify -consistent -split`

  Some of the device pairs are in the Split state, none are in the Consistent state, and the message is:

  ```
  Not All devices in the group 'STAGING' are in 'Consistent, Split'
  states.
  ```

- If the verify command specifies Consistent, Split, Suspended, OR Synchronized states:

```
symrdf -g STAGING -rdfg 129 verify -consistent -split -suspended
-synchronized
```

All device pairs in STAGING are in the Split, Suspended, OR Synchronized state. The following message is displayed, even though NO devices are in the Consistent state:

```
All devices in the group 'STAGING' are in 'Consistent, Split,
Suspended, Synchronized' states.
```

Verify both SRDF mode and pair state in one command line

When verifying both SRDF states and modes in the same command line, Solutions Enabler logically ORs the states, logically ORs the modes, and then logically ANDs the two results.

Example In the following example, a device group named STAGING has devices in:

- Synchronous, and adaptive copy disk modes
- Synchronized, suspended and split states, but NOT consistent state

◆ If the verify command specifies synchronous, OR adaptive copy disk mode, AND Synchronized, Suspended, OR Split states:

```
symrdf -g STAGING -rdfg 129 verify -sync -acp_disk -synchronized
-suspended -split
```

All device pairs in STAGING are using synchronous OR adaptive copy disk mode AND are in the Synchronized, Suspended, OR Split state, and the following message is displayed:

```
All devices in the group 'STAGING' are in 'Synchronized, Suspended,
Split' states and 'Synchronous, Adaptive Copy Disk' modes.
```

◆ If the verify command specifies adaptive copy disk mode AND the Synchronized, Suspended, OR Split state:

```
symrdf -g STAGING -rdfg 129 verify -acp_disk -synchronized -suspended
-split
```

Some device pairs in the STAGING group are using synchronous mode, and the following message is displayed:

```
Not All devices in the group 'STAGING' are in 'Synchronized,
Suspended, Split' states and 'Adaptive Copy Disk' modes.
```

◆ If the verify command specifies synchronous, adaptive copy disk mode AND the Consistent state:

```
symrdf -g STAGING -rdfg 129 verify -sync -acp_disk -consistent
```

None of the device pairs in the STAGING group are in the Consistent state, and the following message is displayed:

```
None of the devices in the group 'STAGING' are in 'Consistent' state
and 'Synchronous, Adaptive Copy Disk' modes.
```

Understanding SRDF pair states and links

IMPORTANT

Before you begin SRDF control operations, you must understand how SRDF devices and links work together to secure data within SRDF configurations.

Note: The following content assumes you understand SRDF devices, including R1, R11, R2, and R21. For a detailed description of SRDF devices, refer to *EMC VMAX3 Family Product Guide.*

An SRDF pair state encompasses:

- SRDF device state on the source (R1) device
- SRDF device state on the target (R2) device
- The number of tracks owed between the R1 and R2 devices (invalid tracks), and
- The SRDF link state between the R1 and R2 devices

Note: See "Invalid tracks in SRDF pairs" on page 44.

Figure 2 shows states SRDF devices and links can report.

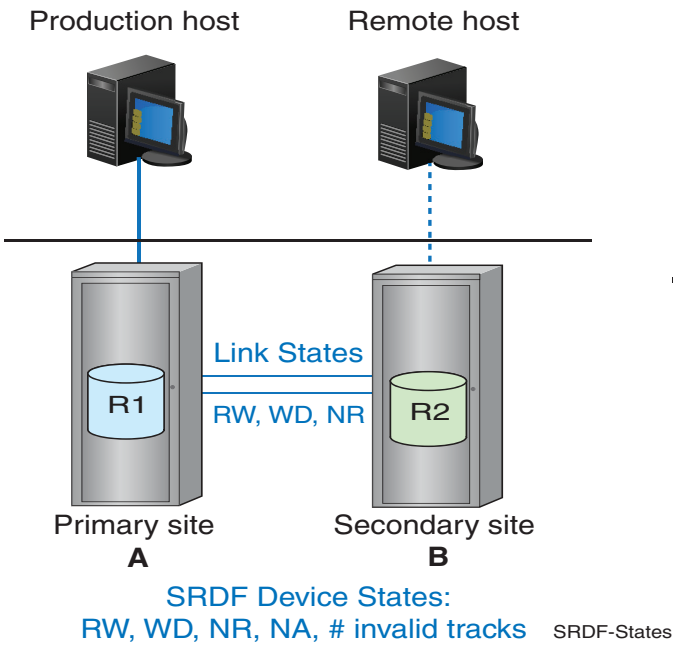


Figure 2 SRDF device and link states

NR	Not Ready. Reads and writes are both disabled.
RW	Ready. Enabled for both reads and writes.
WD	Write Disabled. Enabled for reads but not writes.
NA	Not Available. Unable to report on correct state.

About SRDF pair states

SRDF operation requires that the SRDF pair(s) on which an operation acts must be in a pair state that is valid for that operation.

"SRDF operations and pair states" on page 377 lists control actions and the prerequisite SRDF pair state for each action, including:

- "Cascaded SRDF control operations and applicable pair states" on page 383
- "Concurrent SRDF control operations and applicable pair states" on page 390
- "Consistency group control operations and applicable pair states" on page 396

"Commands to display, query and verify SRDF configurations" on page 34 describes the SYMCLI commands to verify pair states.

Table 7 lists the name and description of SRDF pair states.

Table 7 SRDF pair states (page 1 of 2)

Pair State	Description
SyncInProg	A synchronization is currently in progress between the R1 and the R2 devices: • There are existing invalid tracks between the two pairs, and • The logical links between both sides of an SRDF pair are up.
Synchronized	The R1 and the R2 are currently in a synchronized state: • The same content exists on the R2 as the R1. • There are no invalid tracks between the two pairs.
Split	The R1 and the R2 are currently ready to their hosts, but: • The links are not ready or, • Write disabled.
Failed Over	The R1 is currently not ready or write disabled. Operations have been failed over to the R2.
R1 Updated	The R1 is currently not ready or write disabled to the host: • There are no local invalid tracks on the R1 side, and • The links are ready or write disabled.
R1 UpdInProg	The R1 is currently not ready or write disabled to the host: • There are invalid local (R1) tracks on the source side, • Data is being copied from the R2 to the R1 device, and the links are ready.
Suspended	The SRDF links have been suspended and are not ready or write disabled. If the R1 is ready while the links are suspended, any I/O will accumulate as invalid tracks owed to the R2.
Partitioned	The SYMAPI is currently unable to communicate through the corresponding SRDF path to the remote array. The Partitioned state may apply to devices within an RA group. For example, if SYMAPI is unable to communicate to a remote array from an RA group, devices in that RA group will be marked as being in the Partitioned state. A half pair and a duplicate pair are also reported as Partitioned.
Mixed	A composite SYMAPI device group SRDF pair state. There are different SRDF pair states within a device group.

Table 7 SRDF pair states (page 2 of 2)

Pair State	Description
Invalid	The default state when no other SRDF state applies. • The combination of the R1 device, the R2 device, and the SRDF link states do not match any other pair state. • This state may occur if there is a problem at the disk director level.
Consistent	The R2 SRDF/A capable devices are in a consistent state. Consistent state signifies the normal state of operation for device pairs operating in asynchronous mode.
Transmit Idle	The SRDF/A session cannot push data in the transmit cycle across the link because the link is down.

Invalid tracks in SRDF pairs

On both sides of an SRDF configuration, the array keeps an account of the tracks that are "owed" to the other side. *Invalid tracks* are tracks that are not synchronized between the two devices in an SRDF pair. *Remote invalids* are tracks owed the remote member of the device pair.

For example:

- The logical connection between an R1 device and its R2 is suspended.
- If both devices are made write-accessible, hosts on both sides of the SRDF links write to their respective devices, without the writes being mirrored.
- This creates invalid tracks on the R1 side, and remote invalid tracks on the R2 side.
- Each invalid track represents a track of data that has changed since the two sides were split. To re-establish the logical links between the R1 and R2, the invalid tracks must first be resolved.

How you resolve invalid tracks depends on which control operation you perform. For example if you have remote invalids on both the R1 and R2 sides:

- An `establish` operation copies the modified R1 tracks to the R2 side.

  Any tracks that were modified on the R2 side are overwritten with data from corresponding tracks on the R1 side.
- A `restore` operation copies the modified R2 tracks to the R1 side.

  Any tracks that were modified on the R1 side are overwritten with data from corresponding tracks on the R2 side.

SRDF device and link state combinations

Control actions on an SRDF pair may change the SRDF pair state.

Additionally, the state of a device can change if its front-end or back-end directors change in the SRDF links.

Table 8 shows:

- SRDF pair states resulting from the combination of the states of the source and target devices and the SRDF links.

- The possible R1 or R2 invalid tracks for each SRDF pair state.

Table 8 Possible SRDF device and link state combinations

SRDF pair state	Source (R1) SRDF state	SRDF link state	Target (R2) SRDF state	R1 or R2 invalid tracks
Synchronized	Ready (RW)	Ready (RW)	Not Ready or WD	0
Failed Over	Not Ready or WD	Not Ready	Ready (RW)	—
R1 Updated	Not Ready or WD	Ready (RW) or WD	Ready (RW)	0[1]
R1 UpdInProg	Not Ready or WD	Ready (RW) or WD	Ready (RW)	>0[1]
Split	Ready (RW)	Not Ready or WD	Ready (RW)	—
SyncInProg	Ready (RW)	Ready (RW)	Not Ready or WD	>0
Suspended	Any status[2]	Not Ready or WD	Not Ready or Write Disabled	—
Partitioned[3]	Any status	Not Ready	Not Available	—
Partitioned[4]	Not Available	Not Ready	Any status	—
Mixed	[5]	[5]	[5]	—
Invalid[5]	Any status[6]	Any status	Any status	—
Consistent	Ready (RW)[6]	Ready (RW)	Not Ready or WD	0
Transmit Idle	Ready (RW)[6]	Ready (RW)	Not Ready or WD	—

1. Refers to invalid local (R1) tracks on source.
2. Any status value is possible (Ready, Not Ready, Write Disabled, or Not Available).
3. Viewed from the host locally connected to the source (R1) device
4. Viewed from the host locally connected to the target (R2) device.
5. When no other SRDF states apply, the state defaults to Invalid.
6. The combination of source SRDF, SRDF links, and target SRDF statuses do not match any other SRDF state; therefore, the SRDF state is considered Invalid.

Before you begin

This section includes the following topics:

- "Array access rights"
- "Device external locks"
- "SRDF operations and copy sessions"
- "Migrating R1 to a larger R2 device"
- "Restricting synchronization"
- "Enabling SRDF software and hardware compression"
- "SRDF/A and the consistency exempt option"
- "Enabling mixed-mode workloads on an SRDF director"

- "FAST VP SRDF coordination"

Array access rights

Hosts must have specific access rights to a array to perform certain control operations. Table 9 lists common control operations and the required array access rights.

Table 9 Access rights required by an array

Operations	Required access rights
`symrdf set -rdf`	CFGSYM or SRDF
`symrdf set -rdfa`	CFGSYM or SRDF
`symrdf set rdfa_dse`	CFGSYM or SRDF
`symrdf set rdfa_pace`	CFGSYM or SRDF
`symrdf addgrp`	CFGSYM
`symrdf modifygrp`	CFGSYM
`symrdf removegrp`	CFGSYM
`symqos set IO`	CFGSYM
`symqos reset IO`	CFGSYM

Device external locks

SYMAPI and SYMCLI use device external locks to lock BCV pairs during TimeFinder control operations and to lock SRDF device pairs during SRDF control operations.

When a `symrdf` control command is initiated, device external locks are set on all SRDF devices. Device external locks are automatically released when the control operation completes.

"Managing locked devices" on page 354 describes how to acquire, recover, and release external locks.

SRDF operations and copy sessions

Certain SRDF operations are not allowed for arrays employing either TimeFinder/Snap or TimeFinder/Clone operations, which use copy session pairs. The availability of some SRDF actions depends on the current pair state of the TimeFinder/Snap or TimeFinder/Clone copy session devices.

"SRDF operations and TimeFinder sessions" on page 397 describes the TimeFinder/Snap and TimeFinder/Clone pair states, and which SRDF operations are available in each state.

Migrating R1 to a larger R2 device

You can copy data from an R1 device to a larger R2 device but the following restrictions apply:

- All swap and SRDF/Star operations are blocked.

- The `SYMAPI_RDF_CREATEPAIR_LARGER_R2` option in the options file must be set to `ENABLE`.

  If `SYMAPI_RDF_CREATEPAIR_LARGER_R2` is set to `DISABLE`, all `createpair` operations are blocked.
- Data mirrored to a larger R2 device cannot be restored back to its R1 device.

Note: For some types of file arrays and attached hosts, host-dependent operations may be required to access data migrated to a larger R2 device.

Restricting synchronization

The `SYMAPI_SYNC_DIRECTION` option in the options file allows you to confine SRDF and TimeFinder operations to only establish or restore actions.

HYPERMAX OS

`SYMAPI_SYNC_DIRECTION` option is not supported on VMAX 3 arrays running HYPERMAX OS.

Enginuity 5876 or less

For some sites running Enginuity 5876 or less, it may be desirable to block users on a host from performing either establish or restore operations.

To block a user on a host from executing a restore or an establish action, edit the following entry in the options file:

```
SYMAPI_SYNC_DIRECTION=ESTABLISH | RESTORE | BOTH
```

- `ESTABLISH` restricts operations to only establish actions.
- `RESTORE` restricts operations to restore, failback, and R1 update actions.
- `BOTH` (default) does not restrict any SRDF or TimeFinder actions.

Enabling SRDF software and hardware compression

Compression minimizes the amount of data transmitted over an SRDF link.

You can activate both software and hardware compression simultaneously for SRDF traffic over GigE and Fibre Channel.

Data is first compressed by software and then further compressed by hardware.

Hardware compression is available on Fibre Channel directors.

You can enable software and hardware compression on both the R1 and R2 sides, but the actual compression happens from the side initiating the I/O, make sure you enable compression on the R1 side.

symrdf set rdfg command syntax

To set hardware and software compression for an SRDF group:

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
   [-noprompt] [-i Interval] [-c Count]

.............

set rdfg
    [-hwcomp {on|off}]
    [-swcomp {on|off}]>
```

```
[-both_sides]
```

"Setting SRDF group attributes" on page 91 provides more information about SRDF group attributes.

Examples To turn on software compression on both sides of SRDF group 12:

```
symrdf -sid 134 -rdfg 12 set rdfg -swcomp on -both_sides
```

To turn off hardware compression on both sides of SRDF group 12:

```
symrdf -sid 134 -rdfg 12 set rdfg -hwcomp off -both_sides
```

To view SRDF software and hardware compression status for an SRDF group:

```
symcfg list -rdfg
```

To view software or hardware compression status for a specified group (12) and specified SID (432):

```
symcfg list -sid 432 -rdfg 12
```

SRDF/A and the consistency exempt option

By default, control operations for an active SRDF/A session are targeted at all device pairs in the session.

The `-cons_exempt` option marks devices targeted by the command as consistency exempt. Devices marked consistency exempt can be controlled independently of other devices in the active SRDF/A session.

Enginuity automatically clears the consistency exempt status when:

- The affected device pairs have become consistent, and
- When the data on the R1 gets applied to the R2.

Enabling mixed-mode workloads on an SRDF director

For arrays running Enginuity 5876.x, you can use the `symqos` command to set the percentage of the SRDF director (RA) CPU resources assigned to each workload type.

Workload percentages must add up to 100%, and can include:

- Synchronous I/Os
- Asynchronous I/Os
- Copy I/Os

Workload settings for the director are used until you explicitly reset them. After reset, the array-level distributions are used.

For information on the `symqos` command syntax, see the *EMC Solutions Enabler Array Management CLI User Guide*.

symqos command syntax

The syntax for the command is:

```
symqos -RA -sid SymmID
```

```
   enable -io
   disable -io

symqos -RA -sid SymmID

   set IO -default
       -sync SyncPercent -async AsyncPercent -copy CopyPercent

   set IO -dir <# | ALL>
       -sync SyncPercent -async AsyncPercent -copy CopyPercent

   reset IO -dir <# | ALL>

symqos -RA [-sid SymmID]

      list -io
```

Examples

To enable the workload percentage settings for synchronous, asynchronous, and copy I/Os on SID 1234:

symqos -RA -sid 1234 enable -io

To set the default settings of the workload percentages for all directors on SID 1234 to 60% for Synchronous I/Os, 30% for asynchronous I/Os and 10% for copy I/Os:

symqos -RA -sid 1234 set IO -default -sync 60 -async 30 -copy 10

To set the settings of the workload percentages on director 8G of SID 1234 to 50% for synchronous I/Os, 30% for asynchronous I/Os, and 20% for copy I/Os:

symqos -RA -sid 1234 -dir 8G set IO -sync 50 -async 30 -copy 20

To reset the customized settings of the workload percentages to the default settings on director 8G of SID 1234:

symqos -RA -sid 1234 -dir 8G reset IO

FAST VP SRDF coordination

If *both* arrays on an SRDF link are running Enginuity 5876 or later, you can enable SRDF coordination to instruct FAST VP to factor the R1 device statistics into move decisions on the R2 device.

For information on FAST and FAST VP, see the *EMC Solutions Enabler Array Management CLI User Guide*.

CHAPTER 2
Basic SRDF Control Operations

This chapter describes the following topics:

- Basic SRDF control operations summary 52
  Table 10 on page 52 lists the SRDF control operations described in this chapter, the arguments associated with the symrdf command used to perform the operation, and a summary of the operation.
- SRDF basic control operations 55
  The remainder of this chapter consists of detailed descriptions and command examples for the operations listed in Table 10.

Basic SRDF control operations summary

Table 10 SRDF control operations summary (page 1 of 3)

Control operation	symrdf argument	Description
Set SRDF mode of replication page 55	set mode [sync|asynch|acp_disk|acp_wp|acp|off]	Set the replication mode for a device, device group, composite group, storage group, or list of devices in a device file.
Enable or disable consistency protection page 58	enable disable	Enables or disables consistency protection for SRDF/A capable devices.
Establish an SRDF pair (full) page 58	establish -full	Establishes remote mirroring and initiates a full data copy from the source (R1) device to the target (R2) device. Use this for: • Initial synchronization of SRDF mirrors. • Replacement of a failed drive on the R2 side.
Establish an SRDF pair (incremental) page 60	establish	Establishes remote mirroring and initiates an incremental data copy from the source (R1) device to the target (R2) device. Use this to resynchronize after a split if you can discard the target data.
Failback page 62	failback	Switches data processing from the target side (R2) back to the source (R1) side. Use this to return the source site from the target site after resolving the cause of a failure.
Failover page 64	failover	Switches data processing from the source (R1) side to the target (R2) side. Use this when a failure occurs on the source side.
Invalidate R1 page 65	invalidate r1	Invalidates all tracks on the source (R1) side so that they can be copied over from the target (R2) side.
Invalidate R2 page 65	invalidate r2	Invalidates all tracks on the target (R2) side so that they can be copied over from the source (R1) side.
Make ready the R1 page 67	ready r1	Sets the source (R1) device to be SRDF ready to its local host.
Make ready the R2 page 68	ready r2	Sets the target (R2) device to be SRDF ready to its local host.
Make the R1 not ready page 67	not_ready r1	Sets the source (R1) device to be SRDF not ready to its local host.
Make the R2 not ready page 68	not_ready r2	Sets the target (R2) device to be SRDF not ready to its local host.
Merge the track tables of the R1 and R2 devices page 69	merge	Merges the track tables between the source (R1) and the target (R2) side.
Move one-half of an SRDF pair page 69	half_movepair	Moves one-half of the SRDF device pair to a different SRDF group. **Note:** If the RA ends up supporting more than 64K devices in the new SRDF group, this operation fails.

Table 10 SRDF control operations summary (page 2 of 3)

Control operation	symrdf argument	Description
Move SRDF device pairs page 70	movepair	Moves the SRDF device pair to a different SRDF group. **Note:** If the RA ends up supporting more than 64K devices in the new SRDF group, this operation fails.
Read/write disable target device page 71	rw_disable r2	Read/write disables the target (R2) device to its local host.
Refresh R1 page 71	refresh r1	Marks any changed tracks on the source (R1) side to be refreshed from the R2 side.
Refresh R2 page 71	refresh r2	Marks any changed tracks on the target (R2) side to be refreshed from the R1 side.
Restore from a target device page 72	restore -full	Resumes remote mirroring and initiates a full data copy from the target (R2) device to the source (R1) device. Use this for: • Initial (reverse) synchronization of SRDF mirrors. • Replacement of a failed drive on the R1 side.
Restore from a target device (incremental) page 74	restore	Resumes remote mirroring and initiates an incremental data copy from the target (R2) device to the source (R1) device. Use this for resynchronizing SRDF mirrors after a split if you can discard the source data.
Resume SRDF links page 75	resume	Resumes I/O traffic on the SRDF links for the remotely mirrored SRDF pairs in the group.
Split an SRDF pair page 76	split	Stops remote mirroring between the source (R1) device and the target (R2) device. The target device is made available for local host operations. Use this when both sides require independent access, such as for testing purposes.
Suspend SRDF links page 78	suspend	Suspends I/O traffic on the SRDF links for the remotely mirrored SRDF pairs in the group.
Swap SRDF pairs page 80	swap	Swaps the SRDF personality of the designated dynamic SRDF pair. Source R1 devices become target R2 devices and target R2 devices become source R1 devices.
Swap one-half of an SRDF pair page 79	half_swap	Swaps the SRDF personality of one half of the designated dynamic SRDF pair. Source R1 devices become target R2 devices or target R2 devices become source R1 devices.
Update R1 mirror page 80	update	Updates the source (R1) side with the changes from the target (R2) side while the target (R2) side is still operational to its local hosts. Use this to synchronize the R1 side with the R2 side as much as possible before performing a failback, while the R2 side is still online to the host.
Write disable source device page 82	write_disable r1	Write disables the source (R1) device to its local host.
Write disable target device page 82	write_disable r2	Write disables the target (R2) device to its local host.
Write enable source device page 83	rw_enable r1	Write enables the source (R1) device to its local host.

Table 10 SRDF control operations summary (page 3 of 3)

Control operation	symrdf argument	Description
Write enable target device page 83	rw_enable r2	Write enables the target (R2) device to its local host.

SRDF basic control operations

The remainder of this chapter describes the steps to perform typical SRDF operations.

Set mode

SRDF modes of operation determine the following:

- How R1 devices are remotely mirrored to R2 devices across the SRDF links
- How I/Os are processed in an SRDF solution
- When the production host's write I/O command is acknowledged.

This section describes the commands to set SRDF mode, including:

- "Setting the default mode"
- "Setting the mode"
- "Setting mode: synchronous"
- "Setting mode: adaptive copy"
  - "Setting mode: adaptive copy write pending"
  - "Setting mode: adaptive copy disk"
- "Setting mode: asynchronous"

Setting the default mode

The default mode of operation is adaptive copy disk. If you create device pairs without setting a mode, the devices are created in adaptive copy disk mode.

Use the SYMAPI_DEFAULT_RDF_MODE parameter in the options file to modify the default mode.

Setting the mode

Use the `createpair` command to set the SRDF replication mode when you create SRDF device pairs.

See "symrdf createpair command syntax (-file option)" on page 96.

Use `symrdf set` command to set or modify the SRDF replication mode for a device group, a composite group, a storage group, or for devices listed in a device file.

The syntax to set mode on a device group, composite group, storage group, and device file:

```
symrdf -g DgName set mode Mode
symrdf -cg CgName set mode Mode
symrdf -sg SgName set mode Mode -sid SymmID -rdfg GroupNumber
symrdf -f[ile] FileName set mode Mode -sid SymmID -rdfg GroupNumber
```

Setting mode: synchronous

Synchronous mode maintains a real-time image of data between the R1 and R2 devices.

Data must be successfully stored in cache at both the primary and the secondary site before an acknowledgment is sent to the production host at the primary site.

Synchronous mode ensures that the source (R1) and target (R2) devices contain identical data.

To set the replication mode in group `prod` to synchronous:

```
symrdf -g prod set mode sync
```

Setting mode: adaptive copy

Adaptive copy mode is designed to transfer large amounts of data without loss of performance.

Adaptive copy mode allows the R1 and R2 devices to be more than one I/O out of synchronization. Unlike the asynchronous mode, adaptive copy mode does not guarantee a dependent-write consistent copy of data on R2 devices.

The amount of data (number of tracks) out of synchronization between the R1 and the R2 devices at any given time is determined by the maximum skew value. "Setting adaptive copy disk skew" describes the steps to set the maximum skew value.

Adaptive copy modes revert to the specified mode of operation (synchronous mode or semi-synchronous mode) when certain conditions are met.

There are two types of adaptive copy mode:

- "Setting mode: adaptive copy write pending"
- "Setting mode: adaptive copy disk"

Setting mode: adaptive copy write pending

In adaptive copy write pending (acp_wp) mode, the array acknowledges all writes to the source (R1) device as if it is a local device.

The amount of data (number of tracks) out of synchronization between the R1 and the R2 devices at any given time is determined by the maximum skew value. You can set the maximum skew value using SRDF software.

New data accumulates in cache until it is successfully written to the source (R1) device and the remote director has transferred the write to the target (R2) device.

This SRDF mode is designed to have little or no impact on performance between the host and the array containing the source (R1) device.

HYPERMAX OS

Adaptive copy write pending mode is not supported if the R1 side of the SRDF device pair is on an array running HYPERMAX OS.

You cannot set the mode to adaptive copy write pending mode if the R1 side of an SRDF pair is on an array running HYPERMAX OS.

HYPERMAX OS/Enginuity 5876 backward compatibility

In SRDF configurations where R1 devices are on an array running HYPERMAX OS, connected to one or more arrays are running Enginuity 5876, the following restrictions apply:

- For swap and failover operations - If the R2 is on an array running HYPERMAX OS, and the mode of the R1 is adaptive copy write pending mode, SRDF sets the mode to adaptive copy disk.

- For migrate -replace R1 operations - If the R1 being replaced is on an array running HYPERMAX OS, and the mode of the R1 is adaptive copy write pending mode, SRDF sets the mode of the migrated pair to adaptive copy disk.

Examples To set the replication mode in group `prod to` adaptive copy write pending:

```
symrdf -g prod set mode acp_wp
```

To disable adaptive copy write pending and set the replication mode in group `prod` to synchronous:

```
symrdf -g prod set mode acp_off
```

Setting mode: adaptive copy disk

Adaptive copy disk (acp_disk) mode is designed to transfer large amounts of data without loss of performance.

Because the array cannot fully guard against data loss should a failure occur, EMC recommends:

- Use the adaptive copy disk mode to transfer the bulk of your data to target (R2) devices, and then,
- Switch to synchronous mode to ensure full data protection.

When you set the SRDF mode to adaptive copy disk, the array acknowledges all writes to source (R1) devices as if they were local devices. New data accumulates on the source (R1) device and is marked by the source (R1) side as *invalid tracks* until it is subsequently transferred to the target (R2) device. The remote director transfers each write to the target (R2) device whenever link paths become available.

Examples To set the replication mode in group `prod` to adaptive copy disk:

```
symrdf -g prod set mode acp_disk
```

To disable adaptive copy disk mode and set the replication mode in group `prod` to synchronous:

```
symrdf -g prod set mode acp_off
```

Setting adaptive copy disk skew

Skew is an attribute that defines the maximum number of invalid tracks supported by adaptive copy disk mode.

If the number of invalid tracks defined by the skew attribute is exceeded, the remotely-mirrored device switches to synchronous mode.

As soon as the number of invalid tracks drops below the skew threshold, the remotely-mirrored pair reverts back to adaptive copy mode.

Skew is configured at the device level and may be set to a value between 0 and 65,534 tracks. For devices with more than a 2 GB capacity drive, you can specify a value of 65,535 to indicate all tracks of any given drive.

Examples To:

- Change the adaptive copy skew value to the number of tracks on device `BCV023` of group `prod`, and

- Lock device `BCV023` into adaptive copy disk mode (since the number of invalid tracks cannot exceed the maximum threshold of 65, 535):

```
symrdf -g prod set acp_skew 65535 -bcv BCV023
```

To change the adaptive copy skew value to 30,000 tracks for device `BCV023` of group `prod`:

```
symrdf -g prod set acp_skew 30000 -bcv BCV023
```

Setting mode: asynchronous

In asynchronous mode (SRDF/A), data is transferred from the source (R1) site in predefined timed cycles or *delta sets* to ensure that data at the remote (R2) site is *dependent write consistent*.

The array acknowledges all writes to the source (R1) devices as if they were local devices. Host writes accumulate on the source (R1) side until the cycle time is reached and are then transferred to the target (R2) device in one delta set. Write operations to the target device are confirmed when the current SRDF/A cycle commits the data to disk by successfully de-staging it to the R2 storage devices.

Because the writes are transferred in cycles, any duplicate tracks written to can be eliminated through *ordered write processing*, which transfers only the changed tracks within any single cycle.

The point-in-time copy of the data at the secondary site is slightly behind that on the primary site.

SRDF/A has little or no impact on performance at the primary site as long as the SRDF links contain sufficient bandwidth and the secondary array is capable of accepting the data as quickly as it is being sent across the SRDF links.

When you set the mode as asynchronous for an SRDF group, all devices in the group must operate in that mode.

To set the replication mode in group `prod` to asynchronous:

```
symrdf -g prod set mode async
```

Note: A device status check is performed on all TimeFinder Snap and Clone device pairs in the group before the `set mode async` operation is allowed. Depending on the device pair state, asynchronous mode may not be allowed for devices employing either TimeFinder/Snap or TimeFinder/Clone operations. "SRDF operations and TimeFinder sessions" on page 397 explains the applicable device pair states for TimeFinder/Snap or TimeFinder/Clone operations.

"SRDF/Asynchronous operations" on page 124 provides additional information on operating in asynchronous mode.

Establish (full)

A full establish initiates the following activities for each specified SRDF pair in a device group, consistency group, storage group, or list of devices in a device file:

- The target (R2) device is write disabled to its local host I/O.

- Traffic is suspended on the SRDF links.
- All the tracks on the target (R2) device are marked invalid.
- All tracks on the R2 side are refreshed by the R1 source side. The track tables are merged between the R1 and R2 side.
- Traffic is resumed on the SRDF links.

When the establish control operation has successfully completed and the device pair is in the Synchronized state, the source (R1) device and the target (R2) device contain identical data.

A full establish on SRDF pairs is required only:

- At initial set up of SRDF pairs,
- When an R2 member of an SRDF pair is either fully invalid, or has been replaced.

Figure 3 illustrates establishing an SRDF pair.

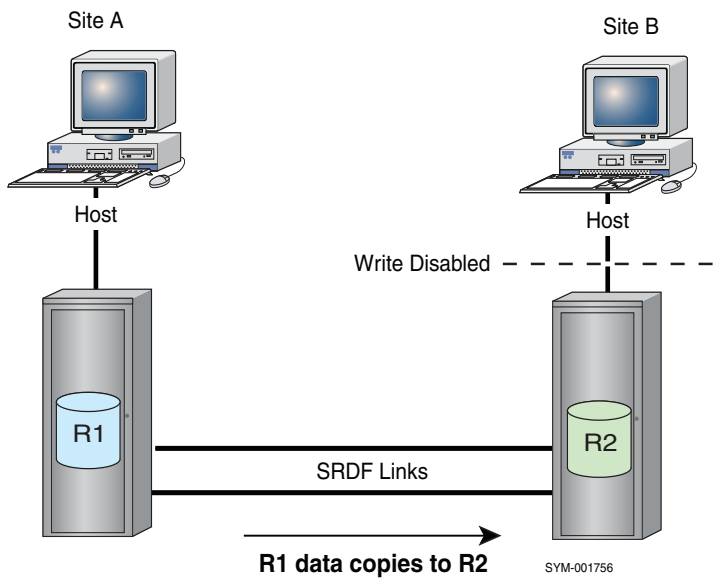


Figure 3 Full establish

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

"establish -full" on page 379 lists the applicable SRDF pair states for this operation.

symrdf establish -full command syntax

You can issue the `establish -full` operation on a device group, composite group, storage group, and device file::

```
symrdf -g DgName establish -full
symrdf -cg CgName establish -full
symrdf -sg SgName establish -full
symrdf -f[ile] FileName establish -full
```

"Creating a device file" on page 94 describes the steps to create a device file.

Use `verify` to confirm that the establish is complete:

- SRDF pairs are in the Synchronized state, and
- Remote mirroring is resumed.

Examples

To establish all the SRDF pairs in the device group `prod`:

```
symrdf -g prod establish -full
```

To establish one SRDF pair with logical device `DEV001` in the device group `prod`:

```
symrdf -g prod establish -full DEV001
```

To establish a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod establish -full DEV001 DEV002 DEV003
```

Note: The R2 may be set to read/write disabled (not ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler CLI Command Reference*.

Establish (incremental)

An incremental establish re-synchronizes data on the source (R1) and target (R2) device when a split RDF pair is rejoined. Only the new data that was updated on the source (R1) device while the SRDF pair was split is copied, greatly reducing the amount of data that is copied.

An incremental establish initiates the following activities for each specified SRDF pair in a device group:

- The target (R2) device is write disabled to its local host I/O.
- Traffic is suspended on the SRDF links.
- The invalid tracks on the target (R2) device are refreshed from the changed tracks of the source (R1) device.
- The track tables are merged between the source (R1) device and the target (R2) device.
- Traffic is resumed on the SRDF links.

The SRDF pair is in the Synchronized state when the source (R1) device and the target (R2) device contain identical data.

Figure 4 illustrates the incremental establishing of an SRDF pair.

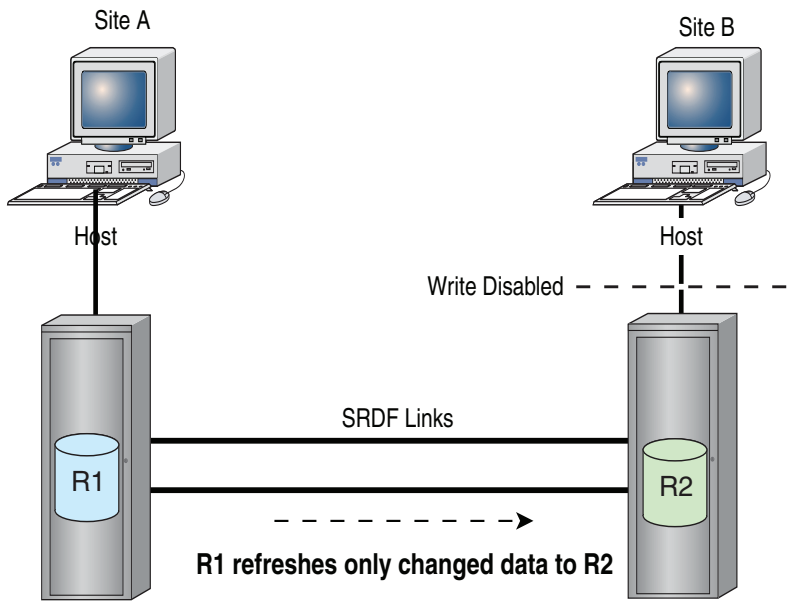


Figure 4 Incremental establish

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

"establish" on page 379 lists the applicable SRDF pair states for this operation.

symrdf establish command syntax

You can issue the `incremental -establish` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName establish
symrdf -cg CgName establish
symrdf -sg SgName establish
symrdf -f[ile] FileName establish
```

"Creating a device file" on page 94 describes the steps to create a device file.

Examples

To initiate an incremental establish on all SRDF pairs in the `prod` device group:

```
symrdf -g prod establish
```

To initiate an incremental establish on one SRDF pair with logical device `DEV001` in the `prod` device group:

```
symrdf -g prod establish DEV001
```

To initiate an incremental establish for a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod establish DEV001 DEV002 DEV003
```

Note: R2 may be set to read/write disabled (not ready) if SYMAPI_RDF_RW_DISABLE_R2=ENABLE is set in the options file. For more information, refer to the *EMC Solutions Enabler CLI Command Reference*.

Failback

After a failover (planned or unplanned), use the failback command to resume normal SRDF operations by initiating read/write operations on the source (R1) devices, and stop read/write operations on the target (R2) devices.

Failback initiates the following activities for each specified SRDF pair in a device group:

- The target (R2) device is write disabled to its local hosts.
- Traffic is suspended on the SRDF links.
- If the target side is operational, and there are invalid remote (R2) tracks on the source side (and the force option is specified), the invalid R1 source tracks are marked to refresh from the target side.
- The invalid tracks on the source (R1) side are refreshed from the target R2 side. The track tables are merged between the R1 and R2 sides.
- Traffic is resumed on the SRDF links.
- The source (R1) device is read/write enabled to its local hosts.

The target (R2) devices become read-only to their local hosts while the source (R1) devices are read/write enabled to their local hosts.

Failback includes the following general steps:

1. Stop I/Os on the failover host at site B.
2. Make all R2 devices in array at site B Not Ready or Read Only (Write Disabled) to the host.
3. If the array at site A was powered off, ensure that SRDF links between array A and array B are disabled before powering on the array at site A.
4. If the array at site A running Enginuity 5874 or earlier was powered off and you do not want to discard its changed data, disconnect or disable the SRDF links between array A and array B before powering on array A. In this way, you will prevent changed data from array B (secondary) from moving automatically to array A (primary).
5. Power on the array at site A and make R1 devices Read/Write enabled to the production host.
6. Enable the SRDF links between the array at site A and the array at site B.
7. Bring the SRDF links online and restart the local host. The R1 devices automatically receive data from the R2 devices which accumulated invalid tracks on their R2 SRDF mirrors during production processing.
8. Once all SRDF pairs are synchronized, enable consistency groups on the SRDF links between the array at site A and the array at site B.
9. Restart the site A host and applications.

Figure 5 shows the failback of an SRDF pair.

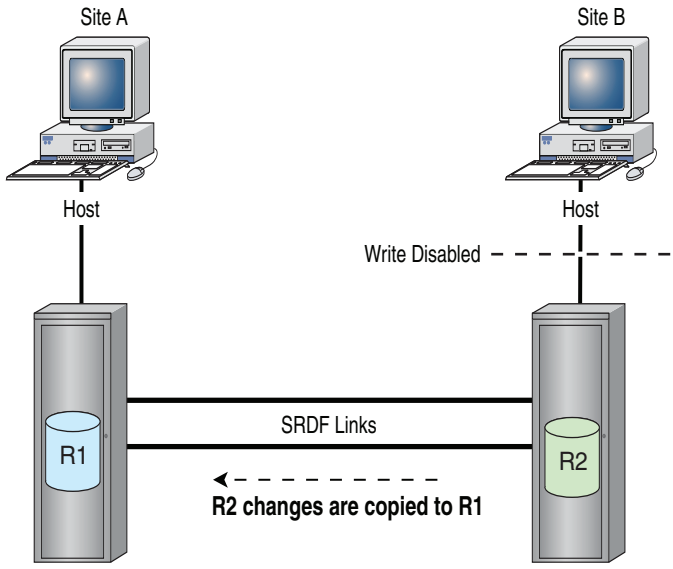


Figure 5 Failback of an SRDF device

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

"failback" on page 379 lists the applicable SRDF pair states for this operation.

symrdf failback command syntax

You can issue the `failback` control operation a device group, composite group, storage group, and device file:

```
symrdf -g DgName failback
symrdf -cg CgName failback
symrdf -sg SgName failback
symrdf -f[ile] FileName failback
```

Note: The R2 may be set to read/write disabled (not ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler CLI Command Reference*.

"Creating a device file" on page 94 describes the steps to create a device file.

Examples To initiate a failback on all the SRDF pairs in the `prod` device group:

```
symrdf -g prod failback
```

To initiate a failback on one SRDF pair, `DEV001`, in the `prod` device group:

```
symrdf -g prod failback DEV001
```

To initiate a failback on a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod failback DEV001 DEV002 DEV003
```

Failover

Failovers are used to move processing to the R2 devices during scheduled maintenance (planned failover) or when an outage makes the R1 devices unreachable (unplanned failover).

A failover transfers processing to the target (R2) devices and makes them read/write enabled to their local hosts.

Failover initiates the following activities for each specified SRDF pair in a device group:

- If the source (R1) device is operational, the SRDF links are suspended.
- If the source side is operational, the source (R1) device is write disabled to its local hosts.
- The target (R2) device is read/write enabled to its local hosts.

A *planned* failover is a controlled failover operation to test the robustness of the disaster restart solution, or to perform maintenance at the primary site. The secondary site temporarily becomes the primary/production site.

A planned failover includes the following general steps:

1. Shut down all applications on the production host.
2. Take all SRDF links between array A and array B offline to suspend remote mirroring.
3. When SRDF/CG is enabled, disable consistency groups between array A and array B.
4. Swap personalities between R1 and R2 devices.

   SRDF devices at array B are now R1 devices.

   SRDF devices at array A are now R2 devices.

   In SRDF/S configurations, devices are ready to resume production operations at array B.
5. When SRDF/CG is used, enable consistency between array B and array A.
6. Bring all SRDF links between array B and array A online to resume remote mirroring.
7. Start production applications from the host attached to array B.

An *unplanned* failover moves production applications from the primary site to the secondary site after an unanticipated outage at the primary site, and the primary site is not available.

A unplanned failover includes the following general steps:

1. Take all SRDF links between array A and array B offline to suspend remote mirroring.
2. Change the R2 device states to Read/Write to the secondary host connected to array B.
3. Start applications on the secondary host and resume production to write-enabled R2 devices in array B.

Figure 6 shows failover of an SRDF pair.

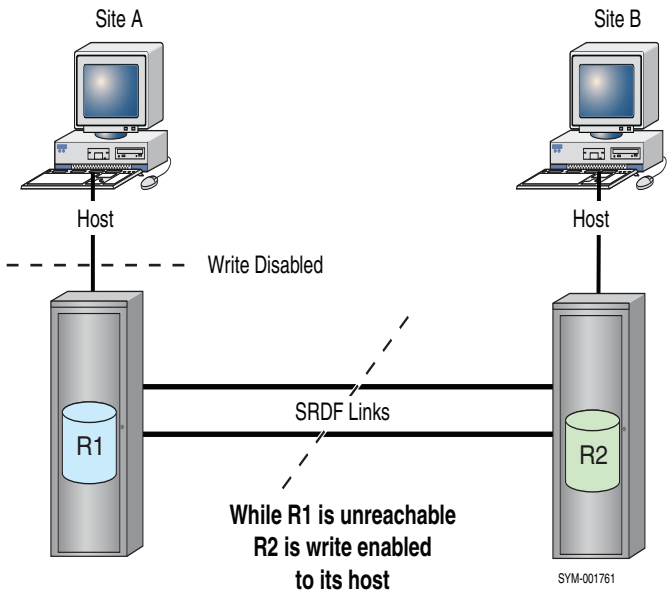


Figure 6 Failover of an SRDF device

"failover" on page 379 lists the applicable SRDF pair states for this operation.

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

---

symrdf failover command syntax

You can issue the `failover` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName failover
symrdf -cg CgName failover
symrdf -sg SgName failover
symrdf -f[ile] FileName failover
```

"Creating a device file" on page 94 describes the steps to create a device file.

Examples To perform a failover on all the pairs in the `prod` device group:

```
symrdf -g prod failover
```

To perform a failover on one SRDF pair with device DEV001 in the prod device group:

```
symrdf -g prod failover DEV001
```

To perform a failover on a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod failover DEV001 DEV002 DEV003
```

Invalidate R1 tracks

The `invalidate r1` action invalidates all tracks on the source (R1) side, so they can be copied over from the target (R2) side.

symrdf invalidate r1 command syntax

You can issue the `invalidate r1` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName invalidate r1
symrdf -cg CgName invalidate r1
symrdf -sg SgName invalidate r1
symrdf -f[ile] FileName invalidate r1
```

Note: Use the `-nowd` option with the `invalidate r1` command to bypass the validation check to ensure that the target of operation is write disabled to the host.

"Creating a device file" on page 94 provides details about defining a device file.

Examples

To invalidate the source (R1) devices in all the SRDF pairs in device group `prod`:

```
symrdf -g prod invalidate r1
```

To invalidate the source (R1) device in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod invalidate r1 DEV007
```

To invalidate the source (R1) device for a list of SRDF pairs in device group `prod`:

```
symrdf -g prod invalidate r1 DEV002 DEV003 DEV007
```

Note: To invoke this operation, the SRDF pairs at the source must already be Suspended and write disabled (not ready).

Invalidate R2 tracks

The `invalidate r2` action invalidates all tracks on the target (R2) side so that they can be copied over from the source (R1) side.

symrdf invalidate r2 command syntax

You can issue the `invalidate r2` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName invalidate r2
symrdf -cg CgName invalidate r2
symrdf -sg SgName invalidate r2
symrdf -f[ile] FileName invalidate r2
```

Note: Use the `-nowd` option with the `invalidate r1` or `invalidate r2` commands to bypass the validation check to ensure that the target of operation is write disabled to the host.

"Creating a device file" on page 94 provides details about defining a device file.

Examples

Note: To invoke this operation, the SRDF pairs at the source must already be Suspended and write disabled (not ready).

To invalidate the target (R2) devices in all the SRDF pairs in device group `prod`:

```
symrdf -g prod invalidate r2
```

To invalidate the target (R2) device in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod invalidate r2 DEV007
```

Make R1 ready

The ready state means the specified mirror is ready to the host. The mirror is enabled for both reads and writes.

`ready r1` sets the source (R1) devices to ready for their local hosts.

This operation is particularly helpful when all SRDF links are lost and the devices are operating in domino mode.

"ready R1" on page 379 lists the applicable SRDF pair states for this operation.

symrdf ready r1 command syntax

You can issue the `ready r1` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName ready r1
symrdf -cg CgName ready r1
symrdf -sg SgName ready r1
symrdf -f[ile] FileName ready r1
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples

To make the source (R1) device ready in all the pairs in device group `prod`:

```
symrdf -g prod ready r1
```

To make the source (R1) device ready in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod ready r1 DEV007
```

To make the source (R1) device ready in a list of SRDF pairs in device group `prod`:

```
symrdf -g prod ready r1 DEV002 DEV003 DEV007
```

Make R1 not ready

The not ready state means the specified mirror is not ready to the host. Both reads and writes are disabled.

`not_ready r1` sets the source (R1) devices to not ready for their local hosts.

"not_ready R1" on page 379 lists the applicable SRDF pair states for this operation.

symrdf not-ready r1 command syntax

You can perform the `not_ready r1` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName not_ready r1
symrdf -cg CgName not_ready r1
symrdf -sg SgName not_ready r1
symrdf -f[ile] FileName not_ready r1
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples To make the source (R1) devices not ready in all the SRDF pairs in device group `prod`:

```
symrdf -g prod not_ready r1
```

To make the source (R1) device not ready in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod not_ready r1 DEV007
```

To make the source (R1) device not ready in a list of SRDF pairs, `DEV007`, in device group `prod`:

```
symrdf -g prod not_ready r1 DEV002 DEV003 DEV007
```

Make R2 ready

The ready state means the specified mirror is ready to the host. The mirror is enabled for both reads and writes.

`ready r2` sets the target (R2) devices to ready for their local hosts.

"ready R2" on page 380 lists the applicable SRDF pair states for this operation.

symrdf ready r2 command syntax

You can issue the `ready r2` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName ready r2
symrdf -cg CgName ready r2
symrdf -sg SgName ready r2
symrdf -f[ile] FileName ready r2
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples To make the target (R2) devices ready in all the SRDF pairs in device group `prod`:

```
symrdf -g prod ready r2
```

To make the target (R2) device ready in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod ready r2 DEV007
```

To make the source (R2) device ready in a list of SRDF pairs in device group `prod`:

```
symrdf -g prod ready r2 DEV002 DEV003 DEV007
```

Make R2 not ready

The not ready state means the specified mirror is not ready to the host. Both reads and writes are disabled.

`not_ready r2` sets the target (R2) devices to not ready for their local hosts.

"not_ready R2" on page 379 lists the applicable SRDF pair states for these operations.

symrdf not-ready r2 command syntax

You can issue the `not_ready r2` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName not_ready r2
```

```
symrdf -cg CgName not_ready r2
symrdf -sg SgName not_ready r2
symrdf -f[ile] FileName not_ready r2
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples To make the target (R2) devices not ready in all pairs in device group `prod`:

```
symrdf -g prod not_ready r2
```

To make the target (R2) device in one pair not ready, `DEV007`, in device group `prod`:

```
symrdf -g prod not_ready r2 DEV007
```

To make the target (R2) device not ready in a list of pairs in device group `prod`:

```
symrdf -g prod not_ready r2 DEV002 DEV003 DEV007
```

Merge track tables

The `merge` operation merges the track tables between the source (R1) and the target (R2) devices.

Merge compares track tables on SRDF device pairs in a device group. Use the `merge` operation to compare the track tables between SRDF device pairs that have been split and re-established.

symrdf merge command syntax

"merge" on page 379 lists the applicable SRDF pair states for this operation.

You can issue the `merge` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName merge
symrdf-cg CgName merge
symrdf -sg SgName merge
symrdf -f[ile] FileName merge
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples To merge the track tables of all the SRDF pairs in device group `prod`:

```
symrdf -g prod merge
```

To merge the track table of one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod merge DEV007
```

To merge the track table of a list SRDF pairs in device group `prod`:

```
symrdf -g prod merge DEV002 DEV003 DEV007
```

Move one-half of an SRDF pair

The `half_movepair` command moves one side of a dynamic SRDF pair from one SRDF group to another. The current invalid track counters on both R1 and R2 stay intact, so resynchronization is required.

Example To move one-half of the SRDF pairing of SRDF group 10 to a new SRDF group 15:

```
symrdf half_movepair -sid 123 -file devicefile -rdfg 10 -new_rdfg 15
```

This command moves the first device listed in each line of the device file to the new SRDF group.

"Moving dynamic SRDF device pairs" on page 108 also provides information on the `symrdf movepair` command for device files.

Move both sides of SRDF device pairs

The `movepair` command moves both the R1 and R2 sides of devices from one SRDF group to another. The current invalid track counters on both R1 and R2 stay intact, so resynchronization is required.

Note: You can only move pairs from like devices: from R1 to R1 or from R2 to R2.

The `movepair` operation has the following restrictions:

- A device cannot move when it is enabled for SRDF consistency.
- A device cannot move if it is in asynchronous mode when an SRDF/A cleanup or restore process is running.
- When moving one mirror of a concurrent R1 or an R21 device to a new SRDF group, the destination SRDF group must not be the same as the one supporting the other SRDF mirror.
- When issuing a full `movepair` operation, the destination SRDF group must connect the same two arrays as the original SRDF group.
- If the destination SRDF group is in asynchronous mode, the SRDF group type of the source and destination group must match. In other words, in asynchronous mode, devices can only be moved from R1 to R1, or from R2 to R2.
- If the destination SRDF group is supporting an active SRDF/A session, the `-cons_exempt` option must be specified.
- If the original SRDF group is supporting an active SRDF/A session, the device pairs being moved must have been suspended using the `-cons_exempt` option.

symrdf movepair command syntax

You can move SRDF pairs using a device file, storage group, or device group:

```
symrdf movepair -sid SID -g DgName -rdfg RDFgroup -new_rdfg NewRDFgroup
symrdf movepair -sid SID -sg SgName -rdfg RDFgroup -new_rdfg NewRDFgroup
symrdf movepair -sid SID -f FileName -rdfg RDFgroup -new_rdfg NewRDFgroup
```

Usage notes

- `-new_rdfg NewRDFgroup` is required.
- `-cons_exempt` allows devices to be moved into an active SRDF/A session without affecting the state of the session or requiring that other devices in the session be suspended.

Example To move pairs in a file from SRDF group 10 to SRDF group 15:

```
symrdf movepair -sid 123 -file devicefile -rdfg 10 -new_rdfg 15
```

This moves the first line of the device file to the new SRDF group, and moves the second device listed in each line of the remote SRDF group that is paired with the new SRDF group.

Read/write disable target device

The `rw_disable r2` action blocks reads from and writes to the target (R2) devices from their local host.

Use this command to set the specified device to the not ready state on the R2 side by making the device not ready on the RA.

"rw_disable R2" on page 380 lists the applicable SRDF pair states for this operation.

symrdf rw_disable r2 command syntax

You can issue the `rw_disable r2` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName rw_disable r2
symrdf -cg CgName rw_disable r2
symrdf -sg SgName rw_disable r2 -rdfg2
symrdf -f[ile] FileName rw_disable r2 -rdfg2
```

"symrdf movepair command syntax" on page 109 provides details on the symrdf movepair command for storage groups and device files.

Examples To read/write disable all the target (R2) mirrors in the SRDF pairs in a device group `prod`:

`symrdf -g prod rw_disable r2`

To read/write disable the target (R2) mirror in the SRDF pair, `DEV007`, in device group `prod`:

`symrdf -g prod rw_disable r2 DEV007`

To read/write disable the target (R2) mirror in a list of SRDF pairs in device group `prod`:

`symrdf -g prod rw_disable r2 DEV002 DEV003 DEV007`

Refresh R1

The refresh R1 mirror action marks any changed tracks on the source (R1) side to refresh from the R2 side.

Use the refresh R1 mirror action when the R2 device holds the valid copy and the R1 device's invalid tracks will be refreshed using the R2 data.

"refresh R1" on page 380 lists the applicable SRDF pair states for this operation.

symrdf refresh r1 command syntax

You can issue the `refresh r1` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName refresh r1
symrdf -cg CgName refresh r1
symrdf -sg SgName refresh r1
symrdf -f[ile] FileName refresh r1
```

Examples To refresh all the source (R1) devices in all the SRDF pairs in the device group `prod`:

`symrdf -g prod refresh r1`

To refresh the source (R1) device in the SRDF pair, `DEV007`, in the device group `prod`:

```
symrdf -g prod refresh r1 DEV007
```

To refresh the source (R1) device in the list of SRDF pairs in the device group `prod`:

```
symrdf -g prod refresh r1 DEV002 DEV003 DEV007
```

Refresh R2

The refresh R2 mirror action marks any changed tracks on the target (R2) side to refresh from the R1 side.

Use the refresh R2 mirror action when the R1 device holds the valid copy and the R2 device's invalid tracks will be refreshed using the R1 data.

"refresh R2" on page 380 lists the applicable SRDF pair states for this operation.

symrdf refresh r2 command syntax

You can issue the `refresh r1` and `refresh r2` operations on a device group, composite group, storage group, and device file:

```
symrdf -g DgName refresh r2
symrdf -cg CgName refresh r2
symrdf -sg SgName refresh r2
symrdf -f[ile] FileName refresh r2
```

Examples

To refresh the target (R2) devices in all the SRDF pairs in device group `prod`:

```
symrdf -g prod refresh r2
```

To refresh the target (R2) device in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod refresh r2 DEV007
```

To refresh the target (R2) device for a list of SRDF pairs in device group `prod`:

```
symrdf -g prod refresh r2 DEV002 DEV003 DEV007
```

Restore (full)

Full restore copies the entire contents of the target (R2) device to the source (R1) device. After the restore operation completes, the pairs synchronize.

When a restore is initiated for each specified SRDF pair in a device group, the following occurs:

- The source (R1) device is write disabled to its local hosts.
- The target (R2) device is write disabled to its local hosts.
- Traffic is suspended on the SRDF links.
- All tracks on the source (R1) device are marked as invalid.
- All R1 tracks are refreshed from the R2 side. The track tables are merged between the R1 and R2 side.
- Traffic is resumed on the SRDF links.

- The source (R1) device is read/write enabled to its local hosts.

The restoration process is complete when the source (R1) and target (R2) device contain identical data. After the restore is complete, the SRDF pair is in the Synchronized state.

Note: R2 may be set to read/write disabled (not ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler CLI Command Reference*.

Figure 7 illustrates restoring an SRDF pair.

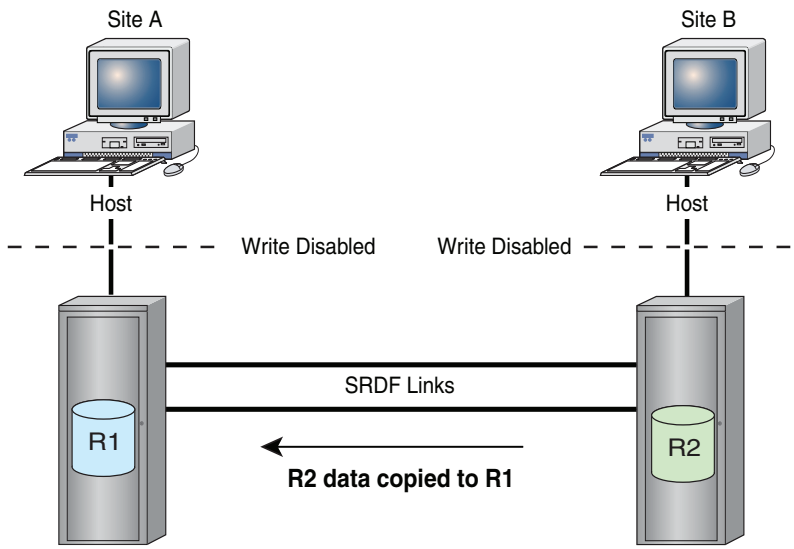


Figure 7 Restore (full) an SRDF device

"restore -full" on page 379 lists the applicable SRDF pair states for this operation.

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

symrdf restore -full command syntax

You can issue the `restore -full` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName restore -full
symrdf -cg CgName restore -full
symrdf -sg SgName restore -full
symrdf -f[ile] FileName restore -full
```

"Creating a device file" on page 94 provides details about defining a device file.

Use `verify` to confirm that the SRDF pairs are in the Synchronized state.

Examples To initiate a full restore on all SRDF pairs in the `prod` device group:

```
symrdf -g prod restore -full
```

To initiate a full restore on one SRDF pair with logical device `DEV001` in the `prod` device group:

```
symrdf -g prod restore -full DEV001
```

To initiate a full restore on a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod restore -full DEV001 DEV002 DEV003
```

Restore (incremental)

An incremental restore re-synchronizes data from the target (R2) to the source (R1) device when a split RDF pair is rejoined. Only those tracks on the target (R2) device that changed while the SRDF pair was split are copied, greatly reducing the amount of data that is copied.

For example, when results from running a new application on the target (R2) device are desirable, and you want to move the changed data and the new application to the source (R1) device.

An incremental restore initiates the following activities for each specified SRDF pair in a device group:

- The source (R1) device is write disabled to its local hosts.
- The target (R2) device is write disabled to its local hosts.
- Traffic is suspended on the SRDF links.
- The invalid tracks on the source (R1) device are refreshed from the changed tracks on the target (R2) side. The track tables are merged between the R1 and R2 side.
- Traffic is resumed on the SRDF links.
- The source (R1) device is read/write enabled to its local hosts.

After the restore control operation has successfully completed, the SRDF pairs will synchronize.

Note: R2 may be set to read/write disabled (not ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler CLI Command Reference*.

Figure 8 illustrates the incremental restore of an SRDF pair.

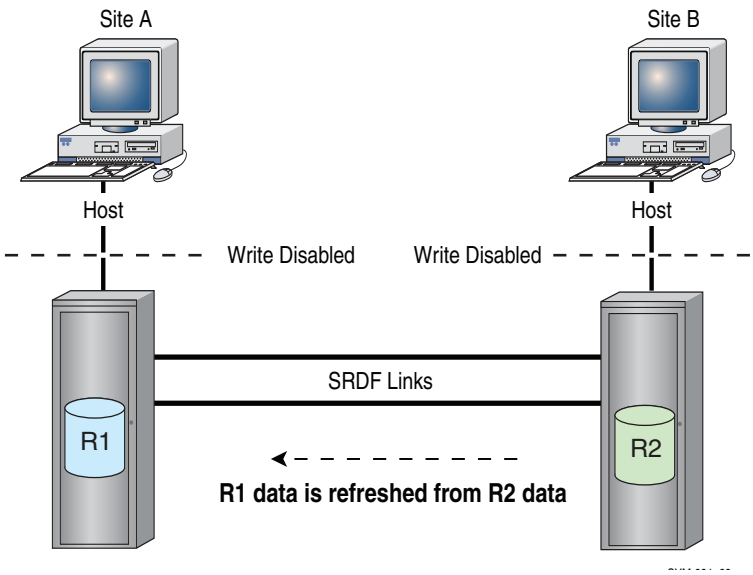


Figure 8 Incremental restore an SRDF device

"restore" on page 379 lists the applicable SRDF pair states for this operation.

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

---

symrdf restore command syntax

You can issue the incremental `restore` operation con a device group, composite group, storage group, and device file:

```
symrdf -g DgName restore
symrdf -cg CgName restore
symrdf -sg SgName restore
symrdf -f[ile] FileName restore
```

"Creating a device file" on page 94 provides details about defining a device file.

Use `verify` to confirm that the SRDF pairs are in the Synchronized state.

Examples To initiate an incremental restore on all SRDF pairs in the `prod` device group:

```
symrdf -g prod restore
```

To initiate an incremental restore on one SRDF pair with logical device `DEV001` in the `prod` device group:

```
symrdf -g prod restore DEV001
```

To initiate an incremental restore for a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod restore DEV001 DEV002 DEV003
```

Resume I/O on links

The `resume` action argument resumes I/O traffic on the SRDF links for all remotely mirrored SRDF pairs in a group or device file.

"resume" on page 380 lists the applicable SRDF pair states for this operation.

symrdf resume command syntax

You can issue the `resume` control operation on device group, composite group, storage group, and device file:

```
symrdf -g DgName resume
symrdf -cg CgName resume
symrdf -sg SgName resume
symrdf -f[ile] FileName resume
```

Note: The `resume` operation is rejected if you do not specify `-force` when the merge track table is required,

Examples

To resume the SRDF links between all the SRDF pairs in device group `prod`:

```
symrdf -g prod resume
```

To resume the SRDF links between one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod resume DEV007
```

To resume the SRDF links (between the pairs) on a list of SRDF pairs in device group `prod`:

```
symrdf -g prod resume DEV002 DEV003 DEV007
```

Split

Split SRDF pairs when you require read and write access to the target (R2) side of one or more devices in a device group, composite group, or device file.

A split performs the following activities for each specified SRDF pair in a device group:

- Traffic is suspended on the SRDF links.
- The target (R2) device is read/write enabled to its local hosts.
- After the target (R2) device is split from the source (R1) device, the SRDF pair is in the Split state.

Figure 9 illustrates splitting an SRDF pair.

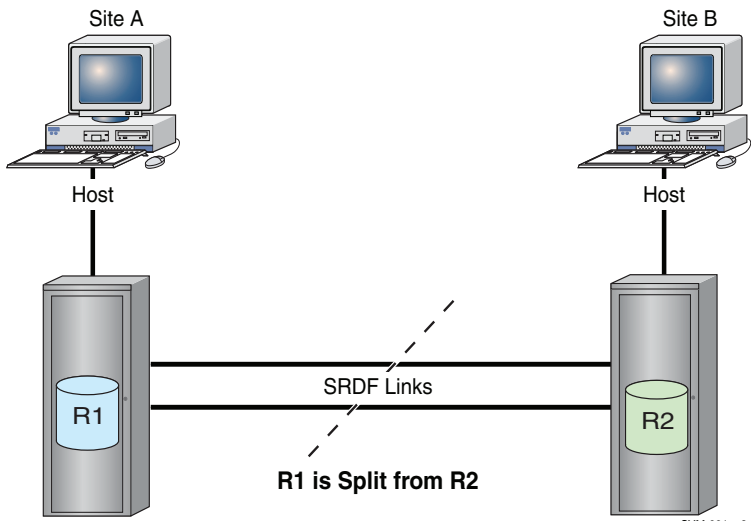


Figure 9 Split an SRDF pair

"split" on page 379 lists the applicable SRDF pair states for this operation.

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to establish. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

symrdf split command syntax

You can issue the `split` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName split
symrdf -cg CgName split
symrdf -sg SgName split
symrdf -f[ile] FileName split
```

Note: The `-force` option is required when the device pairs are in domino mode or adaptive copy mode.

"Creating a device file" on page 94 provides details about defining a device file.

Examples To perform a split on all the SRDF pairs in the `prod` device group:

```
symrdf -g prod split
```

To perform a split on one SRDF pair with logical device `DEV001` in the `prod` group:

```
symrdf -g prod split DEV001
```

To initiate a split to a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod split DEV001 DEV002 DEV003
```

Splits that impact databases

If the `split` action impacts the access integrity of a database, additional actions such as freezing may be necessary. The freeze action suspends writing database updates to disk.

Use the `freeze` action in conjunction with the TimeFinder or `split` operation.

Use the `symioctl` command to invoke I/O control operations to freeze access to a specified relational database or database objects.

Note: For access to the specified database, set SYMCLI_RDB_CONNECT to your username and password.

Freeze access to a database

To freeze all I/O access to a specified relational database:

```
symioctl freeze -type DbType Object Object
```

SQL Server allows some or all databases to be specified. Oracle and Informix allow you to freeze or thaw an entire DB array.

If you have set the connection environment variables, the syntax is:

```
symioctl freeze Object Object
```

To freeze databases HR and Payroll:

```
symioctl freeze HR Payroll
```

Thaw access to a database

Once the freeze action is completed, the split may proceed.

When the split operation completes, use the `symioctl thaw` command to resume full I/O access to the database instance.

To resume I/O access:

```
symioctl thaw
```

Oracle databases: Hot backup control

For Oracle only, you can perform hot backup control on a list of tablespace objects. Hot backup control must be performed before and after a freeze/thaw command.

The steps required to split a group of SRDF pairs follows:

1. Issue the `symioctl begin backup` command.
2. Issue the `symioctl freeze` command.
3. Split the SRDF pairs. This may involve several steps depending on your environment.
4. Issue the `symioctl thaw` command.
5. Issue the `symioctl end backup` command.

Consistency groups

"SRDF consistency group operations" on page 155 describes consistency group split operations.

Suspend I/O on links

The `suspend` action suspends I/O traffic on the SRDF links for all remotely mirrored SRDF pairs in the group or device file.

When the suspend has completed successfully, the devices are suspended on the SRDF links and their link status is set to not ready (NR).

Note: The `suspend` operation is rejected if the specified device is in domino mode.

"suspend" on page 380 lists the applicable SRDF pair states for this operation.

symrdf suspend command syntax

You can issue the `suspend` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName suspend [-immediate | -cons_exempt]
symrdf -cg CgName suspend [-immediate | -cons_exempt]
symrdf -sg SgName suspend [-immediate | -cons_exempt]
symrdf -f[ile] FileName suspend [-immediate | -cons_exempt]
```

Note: Use the consistency exempt (`-cons_exempt`) option to allow devices to be suspended without affecting the state of the SRDF/A session or requiring that other devices in the session be suspended.

Examples

To suspend the SRDF links between all the pairs in device group `prod`:

```
symrdf -g prod suspend
```

To suspend the SRDF links between one pair, `DEV007`, in device group `prod`:

```
symrdf -g prod suspend DEV007
```

To suspend the SRDF links (between the pairs) on a list of pairs in device group `prod`:

```
symrdf -g prod suspend DEV002 DEV003 DEV007
```

Suspend/resume timestamp

Suspend/resume causes SRDF link status to change from read/write to not ready and not ready to read/write. This status information is displayed in the output of the `symdev`, `sympd`, and `symdg show` commands.

Note: The timestamp in the displays is relative to the clock on the host where the command was issued and is reported for each SRDF mirror on both the R1 and R2 mirrors. This timestamp is not associated with the R2 data for SRDF/A.

Swap one-half of an SRDF pair

The `half_swap` command swaps the personality of one half of an SRDF relationship. It changes an R1 mirror to an R2 mirror *or* an R2 mirror to an R1 mirror.

You can swap one half of a designated SRDF pair as specified in a device file, device group, or composite group.

The `half_swap` operation has the following restrictions:

- The R2 device cannot be larger than the R1 device.
- A swap cannot be performed during an active SRDF/A session or when cleanup or restore is running.

- Adaptive copy write pending is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS. If the R2 side is on an array running HYPERMAX OS and the mode of the R1 is adaptive copy write pending, SRDF sets the mode to adaptive copy disk.

Example To swap the R1 designation of the associated BCV RDF1 pairs in device group `prod`, and refresh the data on the current R1 side:

```
symrdf -g Prod -bcv half_swap -refresh R1
```

"Half-swapping dynamic R1/R2 devices" on page 113 describes half-swapping dynamic pairs.

Swap SRDF pairs

The `swap` command swaps the personality of both halves in an SRDF relationship. The source (R1) device becomes the target (R2) device *and* the target (R2) device becomes the source (R1) device.

The `swap` operation has the following restrictions:

- A swap is not allowed if the R1 device (which becomes the R2) is currently a target for a TimeFinder/Snap or TimeFinder/Clone emulation. A device may not have two sources for data (in this case, the R1 and the emulation source). This is not allowed even if the emulation session has already completed copying the data.
- Adaptive copy write pending is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS. If the R2 side is on an array running HYPERMAX OS, and the mode of the R1 is adaptive copy write pending, SRDF sets the mode to adaptive copy disk.

IMPORTANT

The current states of the various devices involved in the SRDF swap must be considered before executing a swap action. Table 12 on page 111 lists which states are legal for this operation.

Example To swap the R1 designation of the associated BCV RDF1 pairs in device group `prod`, and refresh the data on the current R1 side:

```
symrdf -g Prod -bcv swap -refresh R1
```

"Swapping dynamic R1/R2 devices" on page 112 describes swap operations for device files, device groups, or composite groups.

Update R1 mirror

The `update` operation starts an update of the source (R1) side after a failover while the target (R2) side may still be operational to its local hosts.

Use `update` to perform an incremental data copy of only the changed tracks from the target (R2) device to the source (R1) device while the target (R2) device is still Write Enabled to its local host.

An update is initiated for each specified SRDF pair in a device group as follows:

- The SRDF (R1 to R2) links are suspended when the SRDF links are up.

- If there are invalid remote (R2) tracks on the source side and the force option was specified, tracks that were changed on the source devices are marked to refresh from the target side.
- The invalid tracks on the source (R1) side are refreshed from the target R2 side. The track tables are merged between the R1 and R2 sides.
- Traffic is resumed on the SRDF links.

IMPORTANT

If you perform an update while the SRDF pair is Suspended and not ready at the source, the SRDF pair types are in an Invalid state as the update completes. To resolve this condition, use the `rw_enable r1` command to make the SRDF pairs become Synchronized.

When the update has completed successfully, the pairs are in the R1 Updated state.

Figure 10 illustrates the update of an SRDF pair.

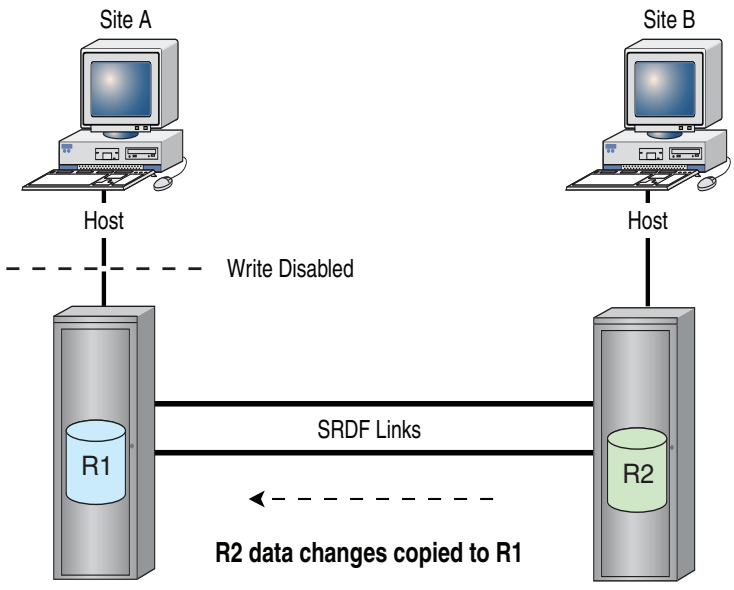


Figure 10 Update SRDF device track tables

"update" on page 379 lists the applicable SRDF pair states for this operation.

Note: When the `symrdf` command is initiated, device external locks are set on all SRDF devices you are about to control. See "Device external locks" on page 46 and "Display devices with a device external lock." on page 35.

symrdf update command syntax

You can issue the `update` operation on a device group, composite group, storage group and device file:

```
symrdf -g DgName update
symrdf -cg CgName update
symrdf -sg SgName update
symrdf -f[ile] FileName update
```

"Creating a device file" on page 94 provides details about defining a device file.

Examples

To initiate an update of all the source (R1) devices in the SRDF pairs, for device group `prod`:

```
symrdf -g prod update
```

To initiate an update of the source (R1) device in the SRDF pair with logical device `DEV001` in device group `prod`:

```
symrdf -g prod update DEV001
```

To initiate an update on a list of SRDF pairs in the device group `prod`:

```
symrdf -g prod update DEV001 DEV002 DEV003
```

Continuous R1 updates

Use the `update -until #` command for scenarios where you want I/O to continue from the remote host and periodically update an inactive R1 device over an extended period of time.

The `-until` option checks the number of invalid tracks that are allowed to build up from the active R2 local I/O before another update (R2 to R1 copy) is triggered. The update sequence loops until the invalid track count is less than the number specified by the `#` value.

Note that these update sequences start with an immediate update once this command is started as follows:

1. Update R1 mirror.
2. Changed tracks are built on R2.
3. Check invalid track count.

Note: If the invalid track count is less than the number of tracks specified by the `-until #` value, the command exits, otherwise, the above sequence of operations for update R1 mirror is retriggered until the threshold is reached.

To update the R1 mirror of device group `prod` continuously until track the number of tracks to be copied is below 1000:

```
symrdf -g prod update -until 1000
```

Write disable R1

The write disable R1 action write disables the source (R1) devices for their local hosts.

"write_disable R1" on page 380 lists the applicable SRDF pair states for these operations.

symrdf write_disable r1 command syntax

You can issue the `write_ disable r1` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName write_disable r1
symrdf -cg CgName write_disable r1
symrdf -sg SgName write_disable r1
symrdf -f[ile] FileName write_disable r1
```

Examples To write disable all the source (R1) mirrors in the SRDF pairs in device group `prod`:

```
symrdf -g prod write_disable r1
```

To write disable the source (R1) mirror in the SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod write_disable r1 DEV007
```

To write disable the source (R1) mirror in a list of SRDF pairs, (`DEV002`, `DEV003`, `DEV007`) in device group `prod`:

```
symrdf -g prod write_disable r1 DEV002 DEV003 DEV007
```

Write disable R2

The write disable R2 action write disables the source (R2) devices for their local hosts.

"write_disable R2" on page 380 lists the applicable SRDF pair states for these operations.

symrdf write_disable r2 command syntax

You can issue the `write_ disable r2` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName write_disable r2
symrdf -cg CgName write_disable r2
symrdf -sg SgName write_disable r2
symrdf -f[ile] FileName write_disable r2
```

Examples To write disable all the target (R2) mirrors in the SRDF pairs in device group prod:

```
symrdf -g prod write_disable r2
```

To write disable the target (R2) mirror in the SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod write_disable r2 DEV007
```

To write disable the target (R2) mirror in a list of SRDF pairs in device group `prod`:

```
symrdf -g prod write_disable r2 DEV002 DEV003 DEV007
```

Write enable R1

The read/write enable R1 action enables the source (R1) devices for their local hosts.

"rw_enable R1" on page 380 lists the applicable SRDF pair states for these operations.

You can issue the `rw_enable r1` operation on a device group, composite group, and device file:

```
symrdf -g DgName rw_enable r1
symrdf -cg CgName rw_enable r1
symrdf -f[ile] FileName rw_enable r1
```

Examples To read/write enable all the source (R1) mirrors in all the SRDF pairs in device group `prod`:

```
symrdf -g prod rw_enable r1
```

To read/write enable the source (R1) mirrors in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod rw_enable r1 DEV007
```

Write enable R2

The read/write enable R2 action write enables the target (R2) devices for their local hosts.

"rw_enable R2" on page 380 lists the applicable SRDF pair states for these operations.

You can issue the `rw_enable r2` operation on a device group, composite group, and device file:

```
symrdf -g DgName rw_enable r2
symrdf -cg CgName rw_enable r2
symrdf -f[ile] FileName rw_enable r2
```

Examples

To read/write enable all the target (R2) mirrors in the SRDF pairs in device group `prod`:

```
symrdf -g prod rw_enable r2
```

To read/write enable the target (R2) mirror in one SRDF pair, `DEV007`, in device group `prod`:

```
symrdf -g prod rw_enable r2 DEV007
```

To read/write enable the target (R2) mirror of a list of SRDF pairs in device group `prod`:

```
symrdf -g prod rw_enable r2 DEV002 DEV003 DEV007
```

CHAPTER 3
Dynamic Operations

This chapter describes the following topics:

- Overview 86
- SRDF group attributes 87
- Adding SRDF groups 88
- Removing SRDF groups 93
- Creating SRDF device pairs 95
- Deleting dynamic SRDF device pairs 105
- Moving dynamic SRDF device pairs 108
- Swapping SRDF devices 110
- Dynamic failover operations 113

Overview

An SRDF group is comprised of SRDF devices and SRDF directors that reside on a given array. The SRDF mirrors that belong to these SRDF devices point to the SRDF partner devices that reside on another array and are configured to the partner SRDF group.

SRDF groups communicate with their partner SRDF groups in another array across the SRDF links. SRDF group configuration parameters include the partner array identification and the set of SRDF directors that belong to the partner SRDF group.

You must create SRDF groups on both ends of the SRDF links.

SRDF groups can be created, modified, and deleted on demand while the array is in operation.

As soon as an empty SRDF group is created on one array, a partner SRDF group must be created on the other array. The SRDF directors assigned to each group share CPU processing power, SRDF ports, and serve all SRDF devices that are added to an SRDF group associated with that director. SRDF directors on each side of the SRDF links cooperate to support regular SRDF I/O operations.

The maximum number of SRDF groups and SRDF groups per SRDF director varies by operating system version:

- Enginuity 5773:
  - 128 SRDF groups,
  - 32 SRDF groups per SRDF director
- Enginuity 5874 and higher:
  - 250 SRDF groups
  - 64 SRDF groups per SRDF director
- HYPERMAX OS
  - 250 SRDF groups per SRDF director

HYPERMAX OS

All SRDF devices and SRDF groups on arrays running HYPERMAX OS are created as dynamic.

For configurations where one array is running HYPERMAX OS, and the second array is running Enginuity 5876, SRDF groups on the 5876 array must be created as dynamic. You cannot pair static SRDF groups or devices on one array with dynamic SRDF groups or devices on a second array.

HYPERMAX OS supports multiple ports per director.

When both arrays connected by an SRDF group are running HYPERMAX OS:

- Up to 250 SRDF groups can be defined across all of the ports per SRDF director or
- Up to 250 SRDF groups can be defined on 1 port on a specific RDF director.

When one array is running HYPERMAX OS and the second array is running Enginuity 5876:

- The port on the array running HYPERMAX OS which is connected to a port on an array running Enginuity 5876 can support up to 64 SRDF groups.

Thus, the maximum number of SRDF groups supported on the HYPERMAX OS director is effectively 186 (250-64).

SRDF group attributes

All SRDF groups have configurable attributes that are applied to the devices in the group, including:

- "Link limbo"
- "Domino mode"
- "Autolink recovery"
- "Hardware compression"
- "Software compression"

Note: SRDF/A device groups have additional configurable attributes. See "Setting SRDF/A group cycle time, priority, and transmit idle" on page 126.

Link limbo

Note: This option is for advanced users only.

Link limbo is an advanced user feature that allows you to set a specific length of time for Enginuity to wait when a down link is detected before updating the link status.

You can specify a link limbo value on the local side or both the local and remote sides of a dynamic SRDF group. If the link status is still not ready after the link limbo time expires, devices are marked not ready to the link.

Valid values for the link limbo timer are 0 through 120 seconds.

The default is 10 seconds.

To protect from session drops after the maximum link limbo time, enable the Transmit Idle feature. "Managing transmit idle" on page 141 provides more information.

IMPORTANT

Because the setting of the link limbo timer affects the application timeout period, it is not recommended to set while running in synchronous mode.
Switching to SRDF/S mode with the link limbo parameter configured for more than 10 seconds could result in an application, database, or host failure if SRDF is restarted in synchronous or semi-synchronous mode.

Domino mode

Under certain conditions, the SRDF devices can be forced into the Not Ready state to the host. For example, if the host I/Os cannot be delivered across the SRDF link.

Use the domino attribute to stop all subsequent write operations to both R1 and R2 devices to avoid data corruption.

While such a shutdown temporarily halts production processing, domino modes can protect data integrity in case of a rolling disaster.

Autolink recovery

If all SRDF links fail, the array remembers the SRDF states of the affected SRDF devices and can automatically restore the devices to these states once the SRDF links become operational.

Enable the Autolink recovery attribute (`-autolink_recovery`) to allow SRDF to automatically restore the SRDF links.

Valid values for `-autolink_recovery` are on (enabled) and off (disabled).

The default is off.

Hardware compression

SRDF hardware compression (`-hwcomp`)is available over Fibre Channel and GigE links. Compression minimizes the amount of data transmitted over an SRDF link.

Valid values for `-hwcomp` are on (enabled) and off (disabled).

The default is off.

Software compression

Software compression (`-swcomp`) can be applied to SRDF traffic over Fibre Channel and GigE SRDF links. If software compression is enabled, Enginuity compresses data before sending it across the SRDF links.

Arrays at both sides of the SRDF links must support software compression and must have the software compression feature enabled in the configuration file.

Valid values for `-swcomp` are on (enabled) and off (disabled).

The default is off.

Managing SRDF groups

This section includes procedures to create, manage, and delete SRDF groups. Topics include:

- "Adding SRDF groups"
- "Setting SRDF group attributes"
- "Adding/removing supporting directors for an SRDF group"
- "Removing SRDF groups"

Adding SRDF groups

The following are requirements for adding an SRDF group:

- The `dynamic_rdf` parameter must be enabled.
- The local or remote array must not be in the symavoid file.

  If either array is in the symavoid file, an error message is generated.
- You can perform multiple operations (`addgrp`, `modifygrp`, `removegrp`), but each operation must complete before the next can be started.

- A group label must be specified when adding a dynamic group.

Create an SRDF group and add pairs

Use the `symrdf addgrp` command to create a new SRDF group.

symrdf addgrp command syntax

```
symrdf addgrp -sid SID -label GrpLabel -rdfg GrpNum [-noprompt] [-i
   Interval] [-c Count]

   ...........

   -dir Dir:Port,Dir:Port,…
   -remote_rdfg GrpNum
   -remote_sid SID
   -remote_dir Dir:Port,Dir:Port,…
   -fibre | -gige | -farpoint
   -link_domino {on|off}
   -remote_link_domino
   -auto_link_recovery {on|off}
   -remote_auto_link_recovery
   -link_limbo Secs
   -rem_link_limbo Secs
```

Where:

- `-dir Dir:Port, Dir:Port` specifies a comma-separated list one or more ports on a local director to be added to the group.
- `-remote_dir Dir:Port, Dir:Port` specifies a comma-separated list one or more ports on a remote director to be added to the group.

HYPERMAX OS

Arrays running HYPERMAX OS support multiple ports per director. You must specify both the director ID and the port number when specifying the local and remote ports to add to the new SRDF group. To specify 3 ports on each array:

```
symrdf addgrp -label new_group -rdfg 39 -remote_rdfg 49
-dir 2f:11,1f:12,2h:3 -remote_dir 1h:2,2e:3,2f:12 -sid 000197100001
-remote_sid 000197100228 -nop
```

Enginuity 5773 - 5876

Arrays running Enginuity 5773 - 5876 support a single port per director. Specify only the director ID when specifying the local and remote ports to add to the new SRDF group. For example:

```
symrdf addgrp -label new_group -rdfg 39 -remote_rdfg 49
-dir 2f -remote_dir 1h -sid 000195700001
-remote_sid 000195700228 -nop
```

Mixed configurations

When one array in an SRDF configuration is running HYPERMAX OS, and one array is running Enginuity 5876, specify only the director ID on the array running 5876, and specify both the director ID and port number on the array running HYPERMAX OS. For example:

```
symrdf addgrp -label new_group -rdfg 39 -remote_rdfg 49
-dir 3h:12 -remote_dir 5f -sid 000197100001
-remote_sid 000195700228 -nop
```

- `-remote_rdfg GrpNum` specifies the SRDF group number on the remote array.
- `-remote_sid SID` specifies the ID of the remote array.
- `-fibre | -gige | -farpoint` species the communication protocol for the group: Fibre Channel, Gigabit Ethernet, or FarPoint.

- `-link_domino {on|off}` enables/disables "Establish (full)".
- `-remote_link_domino {on|off}` enables "Establish (full)" remotely.
- `-auto_link_recovery {on|off}` enables/disables local "Autolink recovery".
- `-remote_auto_link_recovery` enables "Autolink recovery" remotely.
- `-link_limbo` *`0 - 120`* enables "Link limbo".
- `-rem_link_limbo` *`0 - 120`* enables "Link limbo" remotely.

Procedure To create a dynamic SRDF group, do the following:

1. Use the `symcfg list` command to display the arrays visible to the host.
2. Use the `symsan list -sanrdf` command to display the SRDF topology from the local array, including available director pairs on the two arrays.

   For example, to determine which remote directors are visible from array 6180:

   `symsan -sanrdf -sid 6180 -dir all list`

   In this example, the output shows that director 13a on array 6240 is visible from director 12a on array 6180

```
Symmetrix ID: 000194906180

    Flags                Remote
--- ------- ---------------------------------
    Dir Lnk
Dir CT  S   Symmetrix ID Dir WWN
--- --- --- ------------ --- ----------------
12A SO  C   000192606240 13A C465090872090050
14A SO  C   000192602586 15A C465090872016879


Legend:
  Director:
    (C)onfig : S = Fibre-Switched, H = Fibre-Hub
               G = GIGE, - = N/A
    S(T)atus : O = Online, F = Offline, D = Dead, - = N/A

  Link:
    (S)tatus : C = Connected, P = ConnectInProg
               D = Disconnected, I = Incomplete, - = N/A
```

3. Use the `symrdf addgrp` command to create an empty dynamic SRDF group.

   In the following example, the `symrdf addgrp` command:

   - Creates a new dynamic SRDF group, specifying the local array (`-sid 6180)` and remote array (`-remote_sid 6240`).
   - Assigns an SRDF group number for the local array (`-rdfg 4`), and for the remote array (`-remote_rdfg 4`) to the new group.
   - Assigns a group label (`-label dyngrp4`) to the new group.

     This label can be up to 10 characters long, and provides a user-friendly ID to modify or delete the new group.

     The group label is required to add/remove directors from the SRDF group. See "Adding/removing supporting directors for an SRDF group" on page 92.

- Adds directors on the local array (`-dir 12a`) and the remote array (`-remote_dir 13a`) to the new group:

```
symrdf addgrp -sid 6180 -rdfg 4 -label dyngrp4 -dir 12a
-remote_rdfg 4 -remote_sid 6240 -remote_dir 13a
```

IMPORTANT

Network topology is important when choosing director endpoints. If using Fibre Channel protocol, the director endpoints chosen must be able to see each other through the Fibre Channel fabric in order to create the dynamic SRDF links. Ensure that the physical connections between the local RA and remote RA are valid and operational.

4. Use the `symcfg -sid SID list -rdfg GroupNumber` command to confirm that the group was added to both arrays.
5. Use the `symrdf createpair` command to add SRDF pairs to the new group.

Note: When creating an RDF pair between HYPERMAX OS and Enginuity 5876, the maximum symdev number that can be used on the array running HYPERMAX OS is FFBF (65471).

In the following example, the `symrdf createpair` command:

- Adds the dynamic SRDF pairs listed in the device file (`-file dynpairsfile`) to the new dynamic SRDF group 4 (`-rdfg 4`)
- Specifies the local array (`-sid 6180`) as the R1 side for the group (`-type R1`)
- The -invalidate option (`-invalidate R2`) indicates that the R2 devices are the targets that will be refreshed from the R1 source devices.

```
symrdf createpair -sid 6180 -rdfg 4 -file dynpairsfile -type R1
-invalidate R2
```

"Device pairing operations" on page 94 provides more information.

Use the `symcfg list -ra all -switched` command to display all SRDF groups on the local array and its remotely connected arrays.

Setting SRDF group attributes

Use the `symrdf set rdfg` command to set the attributes for an SRDF group.

Note: The remote side must be reachable in order to set the SRDF group attributes.

symrdf set rdfg command syntax

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
   [-noprompt] [-i Interval] [-c Count]

.............

set rdfg
    [-limbo {0 - 120}]
    [-domino {on|off}]
    [-autolink_recovery {on|off}]
```

```
[-hwcomp {on|off}]
[-swcomp {on|off}]
[-both_sides]
```

Where:

- `-both_sides` applies the group attribute to both the source and target sides of an SRDF session. If this option is not specified, attributes are only applied to the source side.
- `-limbo {0 - 120}` enables/disables "Link limbo".
- `-domino {on|off}` enables/disables "Domino mode"
- `-autolink_recovery {on|off}` enables/disables "Autolink recovery"
- `-hwcomp {on|off}]` enables/disables "Hardware compression"
- `-swcomp {on|off}` enables/disables "Software compression"

Note: For arrays running Enginuity 5876 or earlier, you can also use the `symconfigure` command to set SRDF group attributes. For more information, see the *EMC Solutions Enabler Array Management CLI User Guide.*

Examples

To set the link limbo value to one minute (60 seconds) for both sides of SRDF group 4 on array 6180:

```
symrdf -sid 6180 -rdfg 4 set rdfg -limbo 60 -both_sides
```

To set the Link Domino mode on both sides of group 4 on array 6180:

```
symrdf -sid 6180 -rdfg 4 set rdfg -domino on -both_sides
```

To set the Autolink Recovery mode on both sides of group 4 on array 6180:

```
symrdf -sid 6180 -rdfg 4 set rdfg -autolink_recovery on -both_sides
```

To set limbo to thirty seconds and turn off Link Domino and Autolink Recovery modes for SRDF group 12:

```
symrdf -sid 134 -rdfg 12 set rdfg -limbo 30 -domino off -autolink_recovery off
```

To turn on software compression and turn off hardware compression on both sides of the SRDF group 12:

```
symrdf -sid 134 -rdfg 12 set rdfg -swcomp on -hwc off -both_sides
```

Adding/removing supporting directors for an SRDF group

Use the `symrdf modifygrp` command to modify a dynamic SRDF group.

symrdf modifygrp command syntax

```
symrdf modifygrp {-add | -remove} -rdfg GrpNum|-label GrpLabel-sid SID
   -i Interval -c Count -star
   .........
   -dir Dir:Port,Dir:Port,…
   -remote_dir Dir:Port,Dir:Port,…
```

Where:

- `-dir Dir:Port, Dir:Port` specifies a comma-separated list of one or more ports on a local director to be added to the group.

- `-remote_dir Dir:Port, Dir:Port` specifies a comma-separated list of one or more ports on a remote director to be added to the group.

HYPERMAX OS

Arrays running HYPERMAX OS support multiple ports per director. You must specify both the director ID and the port number when modifying the local and remote ports. To add port 12 on local director 3h to SRDF group 38:

```
symrdf modifygrp -add -rdfg 38 -dir 3h:12 -sid 000197100001 -nop
```

Enginuity 5773 - 5876

Arrays running Enginuity 5773 - 5876 support a single port per director. Specify only the director ID when specifying the ports to add/remove to/from the SRDF group. For example:

```
symrdf modifygrp -add -rdfg 38 -dir 3h -sid 000195700001 -nop
```

Mixed configurations

When one array in an SRDF configuration is running HYPERMAX OS, and one array is running Enginuity 5876, specify only the director ID on the array running 5876, and specify both the director ID and port number on the array running HYPERMAX OS. For example:

```
symrdf modifygrp -add -rdfg 38 -dir 3h:12 -remote_dir 5f -sid
000197100001 -remote_sid 000195700228 -nop
```

When adding a director to a dynamic group, the specified director for the local array must be online and a physical link to one online director in the remote array must exist.

IMPORTANT

Making physical cable changes within the SRDF environment may disable the ability to modify and delete dynamic group configurations.

Note: Reassigning directors for SRDF dynamic groups requires that you understand the network fabric topology when choosing director endpoints.

The group label or group number is required for modify operations.

Examples

To remove director 13a from the group `dyngrp4` on the local array 6180:

```
symrdf modifygrp -sid 6180 -label dyngrp4 -remove -dir 13a
```

To add director 12a group `dyngrp4` on the local array 6180:

```
symrdf modifygrp -sid 6180 -label dyngrp4 -add -dir 12a
```

Removing SRDF groups

Use the `symrdf removegrp` command to remove an SRDF group

The following must be true before you can remove an SRDF group:

- Both sides of the SRDF configuration must be defined and reachable
- The group must be empty.

  Use the `symrdf deletepair` command to remove all devices from the group.
- At least one physical connection between arrays must exist.

Note: Deleting the group removes all local and remote director support.

Example In the following example:

- The `symrdf deletepair` command deletes SRDF dynamic pairs defined in a device file `dynpairsfile`,
- The `symrdf removegrp` command removes the local and remote dynamic SRDF groups:

```
symrdf deletepair -sid 80 -rdfg 4 -file dynpairsfile

symrdf removegrp -sid 80 -label dyngrp4
```

Removing an SRDF group from one side of an SRDF configuration

Use the `-symforce` option with `symrdf removegrp` command to remove a dynamic SRDF group from one side of an SRDF configuration.

The following must be true before you can remove one side of an SRDF group:

- The other side is not defined or reachable.

  If the other side of the SRDF configuration is reachable, you cannot execute this command.
- The group is empty.

Example The following example removes `dyngrp4` from array 180 on the local side:

```
symrdf removegrp -sid 180 -label dyngrp4 -symforce
```

Device pairing operations

SRDF enables the creation and deletion of SRDF pairs while the array is in operation.

You can specify the devices to be paired using a device file or storage group.

This section describes the steps to add and delete dynamic SRDF pairs, including:

- "Creating a device file"
- "Creating SRDF device pairs"
- "Creating dynamic concurrent pairs"
- "Deleting dynamic SRDF device pairs"

Creating a device file

To create a device file to define SRDF device pairs:

- Create a text file containing two columns.
- Add a separate line in the file for each device pair.

  All devices for one side of the SRDF pair must be in the first column, and all devices for the other side of the SRDF pair must be in the second column.

  It does not matter which side (R1 or R2) is in which column. The `-type` option of the `symrdf createpair` command determines which column is R1 or R2.

IMPORTANT

All devices for an SRDF side must be in the same column. That is, all R1 devices must be in either the left or right column, and all R2 devices must be in the other column.

HYPERMAX OS Solutions Enabler 8.0.1 or higher/HYPERMAX OS does not support meta-devices.

SRDF device pairs consisting of meta-devices on one side and non-meta-devices on the other side are allowed if the meta-devices are on an array running Enginuity 5876.

Example In the following example, the vi text editor creates the `RDFG148` device file consisting of 7 SRDF pairs for the local and remote arrays.

If the left column is R1, the first line pairs local device 0060 with remote device 0092:

```
vi RDFG148
0060 0092
0061 0093
0062 0094
0063 0095
0064 0096
0065 0097
0066 0098
```

When the `symrdf createpair -file` command references the device file, the `-type` option determines whether the devices in the left column are R1 or R2.

Note: The maximum symdev number that can be used on the HYPERMAX OS array is FFBF (65471).

Creating SRDF device pairs

This section explains the various ways to create dynamic SRDF device pairs. Topic include:

- "Supported devices (thin, standard, diskless)"
- "Blocking createpair when R2 is larger than R1"
- "Verifying host cannot write to target devices with -nowd option"
- "Creating pairs with -file option"
- "Creating pairs with -sg option"
- "Creating pairs with -establish option"
- "Creating pairs with -format option"
- "Creating pairs with -invalidate option"
- "Creating pairs with -restore option"
- "Creating dynamic concurrent pairs"

Supported devices (thin, standard, diskless)

Table 11 lists the valid device type combinations for creating an SRDF pair.

Table 11 Device type combinations for creating SRDF pairs

Device 1	Device 2
Standard	Standard
Thin	Thin
Standard	Diskless[4]
Thin[1]	Diskless[1,4]
Thin[2]	Standard[3]

1. FBA devices require Enginuity 5875 or higher.
   CKD devices are not supported.
2. FBA devices require Enginuity 5875 or higher.
   CKD devices require Enginuity 5876 Q42012 SR or higher.
3. Only on Enginuity versions 5671, 5773.50154, or 5875 and higher.
4. 5876 diskless devices cannot be paired with devices on HYPERMAX OS.

Blocking createpair when R2 is larger than R1

IMPORTANT

R2 devices larger than their corresponding R1 devices cannot restore or failover to the R1.

`SYMAPI_RDF_CREATEPAIR_LARGER_R2` in the options file enables/disables creating SRDF pairs where R2 is larger than its corresponding R1. Valid values for the option are:

`ENABLE` - (default value) `createpair` for devices where R2 is larger than its R1 is allowed.

`DISABLE` - `createpair` for devices where R2 larger than its R1 is blocked.

symrdf createpair command syntax (-file option)

```
symrdf -file Filename -sid SymmID -rdfg GrpNum [-bypass] [-noprompt]
   [-i Interval] [-c Count]
   [-v|-noecho] [-force] [-symforce] [-star]

createpair
   -type {R1|R2}
   -invalidate {R1|R2}
   -establish
   -restore>
   -nowd
   -rp
   -format
   -establish
   -rdf_mode {sync|acp_wp|acp_disk|async}
   -g NewDg
   -remote
   -cons_exempt
```

Where:

- `-type [R1|R2]` specifies whether the devices listed in the left column of the device file are configured as the R1 side or the R2 side.
- `-invalidate [R1|R2]` marks the R1 devices or R2 devices in the list to be the invalidated target for a full device copy once the SRDF pairs are created.
- `-establish` begins copying data to invalidated targets, synchronizing the dynamic SRDF pairs in the device file once the SRDF pairs are created.
- `-restore` begins copying data to the source devices, synchronizing the dynamic SRDF pairs in the device file once the SRDF pairs are created.
- `-nowd` bypasses the check explained in "Verifying host cannot write to target devices with -nowd option" on page 98.
- `-rp` allows the operation even when one or more devices are tagged for RecoverPoint.

  A non-concurrent R1 device can be tagged for RecoverPoint. A RecoverPoint tagged device can be used as an R1 device. A device tagged for RecoverPoint cannot be used as an R2 device (createpair) or swapped to become an R2 device (swap, half-swap).
- `-format` clears all tracks on the R1 and R2 sides to ensure no data exists on either side, and makes the R1 read write to the host.

  You can specify this option with `-establish`, `-type`, `-rdf_mode`, `-cons_exempt`, and `-g`.

  When used with `-establish`, the devices become read write on the SRDF link and are synchronized.
- `-rdf_mode` sets the SRDF mode of the pairs to be one of the following:
  - synchronous (sync),
  - asynchronous (async),
  - adaptive copy disk mode (acp_disk),
  - adaptive copy write pending mode (acp_wp).

  Note: Adaptive copy write pending mode is not supported when the R1 mirror of the RDF pair is on an array running HYPERMAX OS.

  Adaptive Copy Disk is the default mode unless overridden by the SYMAPI_DEFAULT_RDF_MODE options file setting. See "Blocking createpair when R2 is larger than R1" on page 96.
- `-g` specifies the device group name to be created with the devices in the device file.
- `-remote` requests a remote data copy. When the link is ready, data is copied to the SRDF mirror.
- `-cons_exempt` option allows devices to be added without affecting the state of an SRDF/A session or requiring that other devices in the session be suspended.

Verifying host cannot write to target devices with -nowd option

When the SYMAPI_RDF_CHECK_R2_NOT_WRITABLE parameter in the options file is enabled, it verifies that the host cannot write to the R2 devices during `createpair` operations (other than `createpair -invalidate <R1|R2>`). This parameter is disabled by default.

Use the `-nowd` option of the `symrdf createpair` command to bypass this check. The `-nowd` option applies to:

- R2 devices for all `createpair` actions
- R1 devices for the `createpair -invalidate R1`
- The source (R1) devices for `-invalidate R1` and to the target (R2) devices for `-invalidate R2`.

Creating pairs with -file option

"Creating a device file" on page 94 describes the steps to create a device file.

Example In the following example, the `createpair` command:

- Creates device pairs using device pairs listed in a device file `devicefile`,
- Ignores the check to see if the host can write to its targets (`-nowd`),
- Sets the mode to the default (adaptive copy disk) by not specifying another mode:

```
symrdf createpair -sid 123 -file devicefile -type r1 -rdfg 10 -nowd
```

Creating pairs with -sg option

Starting in Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR you can manage SRDF operations using storage groups. "Storage groups" on page 26 provides more information.

Use the -sg *SG_Name* option to create dynamic pairs using storage groups (SGs).

This section describes the following topics:

- "Pairing devices using storage groups"
- "Pairing mixed devices using storage groups"
- "Pairing devices in cascaded storage groups"
- "Pairing devices in storage groups (second hop)"

Pairing devices using storage groups

The createpair operation uses the following logic to pair devices in storage groups:

- R1s are paired to R2s of like sizes. GCM is taken into account.

  SRDF detects whether GCM is set or can be set/unset on local and remote devices. "Geometry Compatible Mode" on page 28 provides more information about GCM.

- If the R2 is larger than R1, the device chosen to be the R2 is as close to the R1 size as possible.
- Device pairs must be the same emulation:
  - CKD 3380 to CKD 3380

- CKD 3390 to CKD 3390
- AS400 512 to AS400 512
- AS400 520 to AS400 520
- FBA to FBA

◆ FBA meta devices are paired as follows:

- Concatenated metas are paired to concatenated metas and striped metas are paired to striped metas.
- The number of members in the two metas must be the same.
- The stripe size of the two metas must be the same.
- Thin-to-thin pairs are created before thin-to-thick pairs.
- Thick-to-thick pairs are created before thin-to-thick pairs.

IMPORTANT

If any of the devices in the two storage groups cannot be paired using these rules, the createpair operation fails.

Example In the following example, storage group localSG includes 4 devices:

```
-------------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------------
000A0   N/A                        TDEV            RW    3278
000A1   N/A                        TDEV            RW    1875
000B1   N/A                        TDEV            RW    4125
000C1   N/A                        TDEV            RW    3278
```

and remote storage group remoteSG also has 4 devices:

```
-------------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------------
00030   N/A                        TDEV            RW    1877
00031   N/A                        TDEV            RW    4125
00050   N/A                        TDEV            RW    3278
00061   N/A                        TDEV            RW    4125
```

The createpair -type r1 operation pairs the devices in the local storage group localSG with devices in remote storage group remoteSG:

```
symrdf createpair -sid 123 -rdfg 456 -sg localSG -type r1 -remote_sg
   remoteSG
```

After the operation, pairings are as follows:

Local storage group		Remote storage group	
Device name	Device size	Device name	Device size
000A0	3278 MB	00050	3278 MB
000A1	1875 MB	00030	1875 MB
000B1	4125 MB	00031	4125 MB
000C1	3278 MB	00061	4125 MB

Pairing mixed devices using storage groups

You can pair devices in a storage group that contains a mixture of RDF and non-RDF devices, or RDF devices with different RDF types, if the remote SG contains devices that can be paired with the R1s in the local SG.

Example In the following example, local storage group localSG contains 4 devices of mixed types. Before the createpair operation, device A0 is an R1 device and B1 is an R2 device:

```
-------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------
000A0   N/A                        RDF1+TDEV       RW    3278
000A1   N/A                        TDEV            RW    1875
000B1   N/A                        RDF2+TDEV       RW    4125
000C1   N/A                        TDEV            RW    3278
```

The createpair operation pairs the devices in the local storage group localSG with devices in remote storage group remoteSG:

- `-sid 123 -sg localSG -type r1` - Create device pairs so that devices in local storage group localSG on array 123 are R1 devices.
- `-remote_sg remoteSG` - Pair the devices in local storage group localSG with devices in remote storage group remoteSG:

```
symrdf createpair -sid 123 -rdfg 456 -sg localSG -type r1 -remote_sg
   remoteSG
```

After the operation, device A0 is an R11 and device B1 is an R21:

```
-------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------
000A0   N/A                        RDF11+TDEV      RW    3278
000A1   N/A                        RDF1+TDEV       RW    1875
000B1   N/A                        RDF21+TDEV      RW    4125
000C1   N/A                        RDF1+TDEV       RW    3278
```

Pairing devices in cascaded storage groups

All combinations of cascaded and non-cascaded storage groups are supported. You can pair all the devices in a parent storage group, or only the devices in a specified child storage group.

To pair all the devices in a local parent storage group, (including devices in any child storage groups) with devices in a remote parent storage group, (including devices in any child storage groups) specify the parent storage group names.

To pair devices in a local child storage group with devices in a specified remote child storage group, specify both child storage groups.

Example To pair devices in local parent storage group SG-P1 (including devices in SG-P1's child storage groups) with devices in remote parent storage group SG-P2 (including devices in SG-P2's child storage groups):

```
symrdf createpair -sg SG-P1 -remote_sg SG-P2
```

Example To pair devices in local child storage group local-SG-Child-1 with devices in remote child storage group remote-SG-Child-2:

```
symrdf createpair –sg local-SG-Child-1 –remote_sg remote-SG-Child-2
```

Pairing devices in storage groups (second hop)

Use the following command:

```
symrdf -sg SgName -sid SID -rdfg GroupNum -remote_sg SgName createpair
   -type {r1|r2} -hop2 -hop2_rdfg GroupNum
```

to pair devices in the local storage group and RDF group with devices in the specified remote storage group and RDF group located at hop 2.

IMPORTANT

To create pairs using the -hop_2 option:
- Devices in the remote storage group must have 2 RDF mirrors and the operation is performed on the other mirror.
- Devices in the remote storage group cannot be R21, R22, or R11 devices before the createpair operation.
- The remote storage group must already exist.

Example The following example creates an R1 -> R21 -> R2 configuration starting with an R1 -> R2 pair.

Before the operation, SG_ABC in RDF group 16 on local SID 085 contains 2 R1 devices:

```
-------------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------------
01AA0   N/A                        RDF1+TDEV       RW    3278
01AB1   N/A                        RDF1+TDEV       RW    4125
```

paired with 2 R2 devices in storage group SG_ABC on remote SID 086 (hop 1):

```
Logical   Sym     T R1 Inv   R2 Inv    K Sym     T...
Device    Dev     E Tracks   Tracks    S Dev     E...
--------------------------------- -- ---------...
N/A       01AA0 RW        0         0 NR 0007A WD...
N/A       01AB1 RW        0         0 NR 0007B WD...
```

On the remote SID 087 (hop 2), storage group SG_ABC_HOP2 in RDF group 6 contains two unpaired devices:

```
-------------------------------------------------------------
```

```
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
-------------------------------------------------------------
0009A   N/A                        TDEV            RW    3278
0009B   N/A                        TDEV            RW    4125
```

The following command:

```
symrdf -sg SG_ABC -sid 085 -rdfg 16 -remote_sg remote_SG_ABC_HOP2
   createpair -type R1 -est -hop2 -hop2_rdfg 6
```

creates an R1 -> R21 -> R2 configuration. The devices at hop 2 (SID 087) become R2 devices:

```
-------------------------------------------------------------
Sym                                Device                Cap
Dev     Pdev Name                  Config         Sts    (MB)
    -------------------------------------------------------------
0009A   N/A                        RDF2+TDEV       RW    3278
0009B   N/A                        RDF2+TDEV       RW    4125
```

The devices at hop 1 that were R2 before the operation, are now R21 devices.

Creating pairs with -establish option

Example In the following example, the `createpair -establish` command:

- Creates device pairs using device pairs listed in a device file `devicefile`,
- Begins copying data to its targets, synchronizing the device pairs listed in the device file.

```
symrdf createpair -file devicefile -sid 55 -rdfg 1 -type R1 -establish
```

Note: For the `createpair -establish` option, the R2 may be set to read/write disabled (not ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler Array Management CLI User Guide*.

Creating pairs with -format option

The format option (`-format`) clears all tracks on the R1 and R2 sides to ensure no data exists on either side, and makes the R1 read write to the host. When you use this option to create dynamic pairs, an application cannot write to these devices.

The `symrdf createpair -format` option has the following restrictions:

- Enginuity 5876 and higher.
- Not supported in concurrent SRDF environments.
- The R1 and R2 cannot be mapped to a host.

Example In the following example, the `createpair -format` command:

- Creates device pairs using device pairs listed in a device file `devicefile`,
- Ignores the check to see if the host can write to its targets (`-nowd`),
- Sets the mode for the device pairs to synchronous (`-rdf_mode sync`)
- Clears tracks on the R1 and R2 sides to ensure no data exists on either side, and makes the R1 read write to the host (`-format`):

```
symrdf createpair -sid 66 -format -file devicefile -type r1 -rdfg 117
   -rdf_mode sync -nop
```

Creating pairs with -invalidate option

Example In the following example, the `symrdf createpair` command:

- Creates new SRDF pairs from the list of device pairs in the file `devicefile`.
- The `-type R1` option identifies the first-column devices in the device file in array 55 as R1 type devices.
- The `-invalidate` option indicates that the R2 devices are the targets to be refreshed from the R1 source devices.
- The `-nowd` option bypasses the validation check to ensure that the target of operation is write disabled to its host.
- The SRDF pairs become members of SRDF group 1.

```
symrdf createpair -sid 55 -file devicefile -rdfg 1 -type R1 -invalidate
   r2 -nowd
```

When the command completes, the pairing information is added to the SYMAPI database file on the host.

When the command completes, you can use the `establish` command to start copying data to the invalidated target devices:

```
symrdf -sid 55 -file devicefile establish -rdfg 1
```

Use the `query` command to check the progress of the establish operation:

```
symrdf -sid 55 -file devicefile query -rdfg 1
```

Once synchronized, you can perform various SRDF operations on SRDF pairs listed in the device file

Creating pairs with -restore option

Example Use the `-restore` option to perform a restore operation to copy data back to the R1 source devices:

```
symrdf createpair -sid 55 -file devicefile -rdfg 1 -type R1 -restore
```

Once the SRDF device pairs are created, the restore operation begins copying data to the source devices, synchronizing the dynamic SRDF device pairs listed in the device file.

The `createpair` with `-restore` operation is rejected when:

- The device is in one of the following BCV pair states: Synchronized, SyncInProg, Restored, RestoreInProg, SplitInProg.
- The device is the source or target of a TimeFinder/Snap operation.
- There is a background BCV split operation in progress.
- Devices are in the backend not ready state.
- There is an optimizer swap in progress on a device.
- The emulation type is not the same (such as, AS/400 has specific pairing rules).
- There are existing local invalid tracks on either the local or remote device.

- The SRDF/A session is active and `-cons_exempt` is not specified.
- The SRDF group is in asynchronous mode and the devices being added are not the same SRDF type R1 or R2.
- The SRDF group is in asynchronous mode with the SRDF links suspended and the `-establish` or `-restore` option is selected.
- The SRDF group is enabled for SRDF consistency protection.
- The operation involves one or more of the following unsupported devices: VCM DB, SFS, RAD, DRV, RAID-S, WORM-enabled devices, 4-way mirror, Meta member.

Creating dynamic concurrent pairs

In concurrent SRDF, R1 devices are mirrored concurrently to two R2 devices that reside in two remote arrays.

Use the `symrdf createpair` command to dynamically create concurrent SRDF pairs. This feature allows a second remote mirror to be dynamically added by converting a dynamic R1 device to a concurrent SRDF device.

Two remote mirrors are supported for any dynamic R1 device. With Enginuity 5875 or higher, both mirrors of a concurrent R1 device can be operating in SRDF/A mode.

"Concurrent Operations" on page 183 provides more information.

The following rules apply when creating a dynamic concurrent SRDF pair:

- The SRDF BCVs designated as dynamic SRDF devices are not supported.
- The two SRDF mirrors of the concurrent device must be assigned to different SRDF groups.
- The concurrent dynamic SRDF, dynamic SRDF, and concurrent SRDF states must be enabled.
- With the `-restore` selection, the `-remote` option is required if the link status for the first created remote mirror is read/write.

To dynamically create a second remote mirror using the `symrdf createpair` command, you must create two separate device files:

- One file containing the first set of R1/R2 device pairs, and
- A second device file listing the same R1 device paired with a different remote R2 device.

In a previous example, the `createpair` command created dynamic device pairs and RDF group 1 using a device file named `devicefile`. Devices in the first column of the device file were configured as R1 devices on array 55:

```
symrdf createpair -file devicefile -sid 55 -rdfg 1 -type R1
```

Use the `createpair` command and specify a second device file to create a second set of device pairs for the pairs created in the previous example.

Example In the following example:

- `symrdf createpair` command creates new SRDF pairs from the list of devices in a second device file `devicefile2`

- `-type R1` tells SRDF that devices listed in the first column of `devicefile2` are R1 type devices on array 55.

  Devices listed in the second-column become the second remote mirror devices.
- `-rdfg 2` configures the new SRDF device pairs as members of SRDF group 2.
- `-invalidate R1|R2` marks the R1 devices and R2 devices to invalidate for a full copy when the SRDF pair is created.

```
symrdf createpair -sid 55 -rdfg 2 -file devicefile2 -type R1
   -invalidate R1|R2
```

Use the `createpair` command with the **`-restore -remote`** options to restore the standard R2 type devices.

Example In the following example:

- `-restore` begins a full copy from the target to the source, synchronizing the dynamic SRDF pairs in the device file.
- `-remote` copies data is copied to the concurrent SRDF mirror when the concurrent link is ready.

  Note: These operations require the remote data copy option, or the concurrent link to be suspended.

```
symrdf createpair -file devicefile2 -sid 55 -type R1 -restore -remote
```

Note: The concurrent mirror device pairs must belong to a separate RA group than those defined in the first device file pairing.

Deleting dynamic SRDF device pairs

This section describes the steps to delete dynamic SRDF pairs. Topics include:

- "Deleting a dynamic SRDF pair"
- "-symforce option to clear local invalid tracks"
- "Deleting one-half of an SRDF pair"

Deleting a dynamic SRDF pair

The `deletepair` operation:

- Cancels the dynamic SRDF pairs,
- Removes the pairing information from the array and the SYMAPI database,
- If the device file option (`-file` *`Filename`*) is specified, changes the specified devices to non-SRDF devices (except for concurrent SRDF pairs).
- If the group option (`-g` *`GroupName`*) is specified, changes the device group to a regular device group (except when an SRDF concurrent pair exists).

  When deleting pairs using the group option:

If additional devices were added to the device group before the `symrdf deletepair` command is issued, those added devices are also changed to non-SRDF devices, and the device group is changed to a regular device group, only if the added devices contained within it were dynamic devices. If the device group contained both SRDF and non-SRDF devices, the device group would be changed to an Invalid state.

The `deletepair` operation is rejected when:

- The device is in one of the following BCV pair states: Synchronized, SyncInProg, Restored, RestoreInProg, and SplitInProg.
- There is a background BCV split operation in progress.
- Devices in the backend are not in the ready state.
- There is an optimizer swap in progress on a device.
- SRDF consistency protection is enabled and the devices were not suspended with the `-cons_exempt` option.
- The SRDF links are not suspended.

IMPORTANT

You must suspend the SRDF links using the `symrdf suspend` command before performing the `symrdf deletepair` command.

Examples To delete pairs for an SRDF group:

- `symrdf suspend` suspends the SRDF links for group NewGrp
- `symrdf deletepair` changes Newgrp to a non-SRDF group

```
symrdf suspend -sid 55 -g NewGrp

symrdf deletepair -sid 55 -g NewGrp
```

To delete pairs using a device file:

- `symrdf suspend` suspends the SRDF links for the devices listed in `devicefile`,
- `symrdf deletepair` deletes the specified SRDF pairs. The devices become non-SRDF devices.
- `-rdfg 2` specifies the SRDF group number:

```
symrdf suspend -sid 55 -file devicefile -rdfg 2

symrdf deletepair -sid 55 -file devicefile -rdfg 2
```

Note: To prevent a device group or a composite group from becoming invalid, first remove the devices from the group before performing the `deletepair` action on a device file.

After execution of the `symrdf deletepair` command, the dynamic SRDF pairs are canceled.

-symforce option to clear local invalid tracks

For Enginuity 5876 and higher, use `-symforce` with `symrdf deletepair` command to:

- Remove the SRDF relationship between the R1 and R2 devices

- Clear any local invalid tracks on these devices.

Example To suspend the SRDF relationship for device pairs listed in device file `devicefile`:

```
symrdf suspend -sid 55 -rdfg 112 -file devicefile
```

To delete the device pairs listed in device file `devicefile`:

```
symrdf deletepair -sid 55 -rdfg 112 -symforce -file devicefile
```

Note: This functionality is not supported for diskless devices and does not delete any device pairs containing R11, R21, or R22 devices.

Deleting one-half of an SRDF pair

The `half_deletepair` command allows you to dynamically remove the SRDF pairing relationship between R1/R2 device pairs. One-half of the specified device pair is converted from an SRDF device to a regular device.

The `half_deletepair` command can be specified using a device file or device group. When specified using a device file, all devices listed in the first column of the file are converted to regular devices (non-SRDF).

"half_deletepair" on page 378 lists the applicable SRDF pair states for `half_deletepair` operations.

The `symrdf half_deletepair` is rejected when:

- The device is in one of the following BCV pair states: Synchronized, SyncInProg, Restored, RestoreInProg, and SplitInProg.
- There is a background BCV split operation in progress.
- Devices in the backend are not in the ready state.
- There is an optimizer swap in progress on a device.
- SRDF consistency protection is enabled and the devices were not suspended with the `-cons_exempt` option.
- The SRDF links are not suspended.

Examples To remove the SRDF pairing from device group `Prod` and convert the devices assigned to `Prod` to regular (non-SRDF) devices, leaving their remote partners as SRDF devices:

```
symrdf suspend -g Prod

symrdf -g Prod half_deletepair
```

To remove the SRDF pairing of SRDF group 4 on array 1123 and convert one-half of those device pairs to regular (non-SRDF) devices:

```
symrdf suspend -sid 123 -rdfg 4 -file devicefile

symrdf half_deletepair -sid 123 -rdfg 4 -file devicefile
```

Note: Use the `symrdf list -half_pair` command to list all half pair devices for a specified SID or SRDF group. In addition to `half_deletepair` operations, half pairs can result from `symrdf failover` operations or a configuration changes.

Grouping, moving and swapping dynamic devices

This section describes the steps to group, move and swap dynamic SRDF devices, including:

- "Grouping dynamic pairs with a device file"
- "Moving dynamic SRDF device pairs"
- "Swapping SRDF devices"

Grouping dynamic pairs with a device file

Device groups are the primary method to manage SRDF devices.

An SRDF device file to allows you to manage the devices specified in the file as a single entity.

Procedure To create a device group:

1. Create a list of device pairings in a device file.
2. Use the `createpair` command to create the dynamic SRDF pairs,
3. Use the `-g` *`GroupName`* option to add the devices in the device file to a device group with the specified name.

   For example, to create dynamic devices as specified in file `devicefile` and add them to a group named Newgrp:

   ```
   symrdf createpair -sid 55 -rdfg 2 -file devicefile -type rdf1
   -invalidate r2 -g NewGrp
   ```

   All SRDF commands for these dynamic pairs can now be executed within the context of the `NewGrp` device group

4. Use the `-g` *`GroupName`* option to SRDF commands to perform operations on all the dynamic SRDF pairs in the group.

   For example, establish the group:

   ```
   symrdf -g NewGrp establish
   ```

Moving dynamic SRDF device pairs

Prior to Enginuity version 5773 SRDF pairs could not be moved between groups. To move pairs, original SRDF pairs had to be deleted, new pairs created, and then moved to a new group.

Beginning with Enginuity 5773, you can move dynamic SRDF devices from one SRDF group to another.

Note: There is no need to fully resynchronize the devices when performing the move. The current invalid track counters on both R1 and R2 stay intact.

This section describes moving dynamic SRDF pairs, including:

- "Moving SRDF pairs"
- "Moving half pairs"

- "SRDF mode after a movepair"

Moving SRDF pairs

Use the `movepair -new_rdfg GrpNum` command to move SRDF pairs.

For SRDF/A sessions, use the consistency exempt (`-cons_exempt`) option to allow devices to be moved into an active SRDF/A session without affecting the state of the session or requiring that other devices in the session be suspended.

To move devices out of an active SRDF/A session without affecting the state of the session, first suspend the devices using the `-cons_exempt` option.

After a successful move, the pair state is unchanged.

"movepair" on page 378 lists the applicable SRDF pair states for movepair operations.

symrdf movepair command syntax

SRDF pairs can be moved using a device file, storage group, or device group:

```
symrdf -file Filename -sid SymmID -rdfg GrpNum movepair -new_rdfg
   GrpNum

symrdf -sg SgName -sid SymmID -rdfg GrpNum movepair -new_rdfg GrpNum

symrdf -g GroupName movepair -new_rdfg GrpNum
```

Note: The `-new_rdfg GrpNum` option is required.

See "Move both sides of SRDF device pairs" on page 70 for an example.

The `movepair` operation has the following restrictions:

- A device cannot move when it is enabled for SRDF consistency.
- A device cannot move if it is in asynchronous mode when an SRDF/A cleanup or restore process is running.
- When moving one mirror of a concurrent R1 or an R21 device to a new SRDF group, the destination SRDF group must not be the same as the one supporting the other SRDF mirror.
- When issuing a full `movepair` operation, the destination SRDF group must connect the same two arrays as the original SRDF group.
- If the destination SRDF group is in asynchronous mode, the SRDF group type of the source and destination groups must match. In other words, in asynchronous mode, devices can only be moved from R1 to R1, or from R2 to R2.
- If the destination SRDF group is supporting an active SRDF/A session, the `-cons_exempt` option must be specified.
- If the original SRDF group is supporting an active SRDF/A session, the device pairs being moved must have been suspended using the `-cons_exempt` option.

Moving half pairs

The `half_movepair` operation moves only one side of an SRDF pair from one SRDF group to another.

See "Move one-half of an SRDF pair" on page 69 for an example.

"half_movepair" on page 379 lists the applicable SRDF pair states for control operations.

After a successful `half_movepair` the pair state can go from partitioned to a different state or vice versa.

For example, when a `half_movepair` action results in a normal SRDF pair configuration, the resulting SRDF pair state will be Split, Suspended, FailedOver or Partitioned.

SRDF mode after a movepair

After a `movepair` or `half_movepair` action, the resulting SRDF mode for the moved device is as follows:

- When moving a device *to* an SRDF group that is currently in asynchronous mode, the resulting SRDF mode for the moved device is asynchronous.
- When moving a device *from* an SRDF group that is in asynchronous mode to an SRDF group that is not in asynchronous mode, the resulting SRDF mode for the moved device will be adaptive copy disk.

Swapping SRDF devices

This section describes the following topics:

- "Required states before a swap operation"
- "Displaying SRDF swap-capable devices"
- "Impact on I/O"
- "Disabling SYMAPI control parameter"
- "Refreshing the data status"
- "Swapping dynamic R1/R2 devices"
- "Half-swapping dynamic R1/R2 devices"
- "Swapping cascaded SRDF devices"

The following general steps are required to perform an R1/R2 personality swap and resume SRDF operations:

- Suspend the SRDF remote mirroring.
- Perform a personality swap by converting the R1 to R2 and the R2 to R1 devices.
- Determine the synchronization direction and synchronize the R1 and the R2 devices.
- Resume remote mirroring.

  Host I/Os are accepted at the secondary site (now R1 device) and are remotely mirrored to the R2 device at the primary site.

Required states before a swap operation

The current states of the various devices involved in the SRDF swap must be considered *before* executing a swap action. Table 12 lists which states are legal for this operation.

Table 12 SRDF device states before swap operation

SRDF state	Source R2 invalids	Target R2 invalids	State after swap
Suspended with R1 Write Disabled	Refresh R1|R2	Refresh R1|R2	Suspended
R1 Updated	Refresh R1	NA	Suspended
Failed Over	Refresh R1	NA	Suspended

Displaying SRDF swap-capable devices

Use the `symrdf list` command with the `-dynamic` option to display SRDF devices configured as dynamic SRDF-capable:

```
symrdf list -dynamic [-R1] [-R2] [-both]
```

- Use the command with no options to display all SRDF-capable devices.
- Use the **-R1** option to display all dynamic SRDF-capable devices that are configured as capable of becoming R1.
- Use the **-R2** option to display all dynamic SRDF-capable devices that are configured as capable of becoming R2.
- Use the **-both** option to display a list of dynamic SRDF-capable devices that are configured as capable of becoming R1 or R2

From the displayed list, determine which dynamic devices you want to swap.

Impact on I/O

When swapping source and target attributes I/O is not allowed to the R1 device, but I/O is allowed to the R2 device.

Disabling SYMAPI control parameter

In the options file, `SYMAPI_CTRL_OF_NONVISIBLE_DEVS` must be set to `DISABLE` to prevent control of devices that are not mapped to the user host. The default is `ENABLE`.

Once devices are dynamically swapped, an `incremental establish` operation is initiated and the devices become immediately available on the link. "Establish (incremental)" on page 60 explains this operation.

Refreshing the data status

Swapping the R1/R2 designation of the SRDF devices can impact the state of your stored data.

The refresh action indicates which device does **not** hold a valid copy of the data before the swap operation begins. If you determine that the R1 holds the valid copy, the action of `refresh R2` will obtain a count of the tracks that are different on the R2 and mark those tracks to refresh from the R1 to the R2 device.

The result will be the reverse if you choose to `refresh R1` as the option:

- `-refresh R1` — The R2 device holds the valid copy and the R1 device's invalid tracks are updated using the R2 data.
- `-refresh R2` — The R1 device holds the valid copy and the R2 device's invalid tracks are updated using the R1 data.

symrdf swap -refresh command syntax

You can issue the `swap` command for device groups, composite groups, storage groups, and device files:

```
symrdf [-g DgName |-cg CgName |-sg SgName |-f FileName]
   [-v | -noecho]
   [-force]
   [-symforce]
   [-bypass]
   [-noprompt]
   [-i Interval]
   [-c Count]
   [-hop2 | -bcv [-hop2] | -all | -rbcv | -brbcv]
   [-rdfg GrpNum]
   [-star]
   [-sid SID]
```

Note: -sid *SID* is required for -sg and -f operations.

Where:

- `-bcv` targets just the BCV devices associated with the SRDF device group for the `swap` action.
  - Use `-all` to target both BCV and standard devices.
  - Use nothing to target only standard devices.
- `-refresh` marks the source R1 devices or the target R2 devices to refresh from the remote mirror.
- `-hop2` targets the SRDF action at the group's second-hop devices in a cascaded SRDF relationship.
  - Use alone to target standard devices
  - Use `-bcv -hop2` to target BCV devices

Example The following example:

- Swaps the R1 designation of the associated BCV RDF1 devices within device group `ProdGrpB`.
- Marks to refresh any modified data on the current R1 side of these BCVs from their R2 mirrors:

```
symrdf -g ProdGrpB -bcv swap -refresh R1
```

Swapping dynamic R1/R2 devices

Dynamic R1/R2 swaps switch the SRDF personality of the SRDF device or composite group. Swaps can also be performed on devices in SRDF/A mode. Dynamic SRDF must be enabled to perform this operation.

With a dynamic swap, source R1 devices become target R2 devices and target R2 devices become source R1 devices.

Dynamic SRDF devices are configured as one of three types: RDF1 capable, RDF2 capable, or both. Devices must be configured as both in order to participate in a dynamic swap.

Dynamic swap operations have the following restrictions:

- Dynamic swap is not supported on arrays where the R2 device is larger than the R1 device.
- Do not perform a dynamic swap on SRDF/A devices enabled for consistency protection or if the SRDF/A session is actively copying.

HYPERMAX OS

- Adaptive copy write pending is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS. If the R2 side is on an array running HYPERMAX OS and the mode of the R1 is adaptive copy write pending, SRDF sets the mode to adaptive copy disk.

Half-swapping dynamic R1/R2 devices

You can issue a `half_swap` command that swaps one half of an SRDF relationship. This command changes an R1 mirror to an R2 mirror or an R2 mirror to an R1 mirror.

The `half_swap operation` has the following restrictions:

- The R2 device cannot be larger than the R1 device.
- A swap cannot be performed during an active SRDF/A session or when cleanup or restore is running.

Swapping cascaded SRDF devices

Swapping of an R21 device in a cascaded SRDF relationship is allowed as long as the R21 device is converted into a concurrent R1 (R11) device.

You can convert a concurrent R1 device into an R21 device.

For example, in an R2->R11->R2 configuration, you can swap either side of the relationship:

- Swap R2-> to get R1-> R21->R2
- Swap R11-> R2 to get R2-> R21->R1

The following swap is allowed:

Swap R1->R21 to get R2-> R11-> R2

The following swap is **not** allowed:

Swap R21->R2 to get R1->R22-> R1

Dynamic failover operations

SRDF dynamic devices can be quickly failed over, swapped, and then re-established all within a single command-line operation.

Note: This functionality requires that dynamic devices be both RDF1 and RDF2 capable.

Dynamic failover establish

The `symrdf failover -establish` command can be used as a composite operation on dynamic SRDF devices to quickly perform the following operations on SRDF devices in the specified group using a single command:

1. Failover the devices in the group.

   R2 devices in the group are made read/write enabled to their local hosts.

   "Failover" on page 64 provides a detailed explanation of a failover operation.

2. After the failover operation has completed, swap the SRDF pair personalities.

   R1 devices become R2 devices and the R2 devices become R1 devices).

   "Swapping dynamic R1/R2 devices" on page 112 provides a detailed explanation with restrictions that apply when performing a dynamic swap operation.

3. Once the devices are dynamically swapped, perform an incremental establish operation.

   The devices become immediately available on the link.

   "Establish (incremental)" on page 60 explains this operation.

The `failover restore` operation has the following restrictions:

- Both the R1 and the R2 devices in the failover must be dynamic SRDF devices.
- The R2 device cannot be larger than its R1 device.
- The swap cannot result in a cascaded R21<-->R21 device pair.
- Cannot execute this command on both mirrors of a concurrent R1 device (composite group operation). This swap would convert the concurrent R1 into a concurrent R2, with a restore on both mirrors of that concurrent R2.

Dynamic failover restore

IMPORTANT

The `symrdf failover -establish` operation does not support devices operating in asynchronous mode with a read/write link because the R2 data is two cycle switches behind the R1 data, and swapping these devices would result in data loss.

Restrictions

- If an SRDF group being failed over is operating in asynchronous mode, then all devices in the group must be failed over in the same operation.
- The R1 and the R2 devices in the failover must be dynamic SRDF devices.
- The R2 device cannot be larger than its R1 device.
- The SRDF swap cannot result in a cascaded R21<-->R21 device pair.
- Not supported by any device group operations with more than one SRDF group.
- Cannot execute this command on both mirrors of a concurrent R2 device (composite group operation). This swap would convert the concurrent R2 into a concurrent R1, with a restore on both mirrors of that concurrent R1.

The `symrdf failover -restore` swaps the R1 and R2 and restores the invalid tracks on the new R2 side (formerly R1) to the new R1 side (formerly R2).

You can execute this command for device groups, composite groups, storage groups, and device files. The devices in this failover can be using synchronous or asynchronous links.

symrdf failover command syntax

For device groups:

```
symrdf -g [-g DgName |-cg CgName |-sg SgName |-f FileName]
   [-bypass]
   [-noprompt]
   [-i Interval]
   [-c Count]
   [-hop2 | -bcv [-hop2] | -all | -rbcv | -brbcv]
   [-rdfg GrpNum]
   [-star]
   [-sid SID]

failover [- immediate | -establish | -restore [-remote]]
```

Note: -sid *SID* is required for -sg and -f operations.

Where:

- `-immediate` - deactivates the SRDF/A session immediately, without waiting for the two cycle switches to complete before starting the `failover -restore` operation.
- `-restore` - causes the dynamic SRDF device pairs to swap personality and start an incremental restore.
- `-remote` - requests a remote data copy flag with failback, failover, restore, update, and resume. When the concurrent link is ready, data is copied to the concurrent SRDF mirror. These operations require the remote data copy option, or the concurrent link to be suspended.

Usage notes

- Enginuity 5773 and higher is required to perform a swap resulting in a cascaded R21 device.
- Enginuity 5773.150 or higher is required to perform a swap resulting in a concurrent R2 device.

CHAPTER 4
SRDF/Asynchronous Operations

This chapter describes the following topics:

- Overview 118
- Transitioning to asynchronous mode 125
- Setting SRDF/A group cycle time, priority, and transmit idle 126
- Enabling/disabling consistency for SRDF/A devices 128
- Adding/removing devices using consistency exempt 129
- Managing Delta Set Extension 133
- Managing transmit idle 141
- Managing SRDF/A write pacing 141
- Displaying SRDF/A 148

Overview

SRDF/Asynchronous (SRDF/A) is a long distance disaster restart solution with fast application response times.

SRDF/A cycle modes

SRDF/A provides an R2 copy that is slightly behind its associated R1. Host writes are collected for a configurable interval (`-cycle_time`) into delta sets. Delta sets are transferred to the remote array in predefined timed cycles.

Control of SRDF/A cycles varies depending on whether the array is running in legacy mode (Enginuity 5773 - 5876) or multi-cycle mode (HYPERMAX OS):

Enginuity 5876

If either array in the solution is running Enginuity 5876, there are 2 cycles on the R1 side, and 2 cycles on the R2 side.

Each cycle switch moves the delta set to the next cycle in the process. This mode is referred to as “legacy mode”.

A new capture cycle cannot start until the transmit cycle completes its commit of data from the R1 side to the R2 side, and the R2 apply cycle is empty.

The basic steps in the life of a delta set in legacy mode include:

1. On the R1 side, host writes collect in the Capture cycle's delta set for a specified number of seconds.

   The length of the cycle is specified using the -cycle_time attribute.

   If a given track is overwritten multiple times, only the last write is preserved.

2. Once the cycle timer expires, and both the R1's Transmit cycle and the R2's Apply cycle are empty:

   - The delta set in the R2's Receive cycle is moved to the R2's Apply cycle, from which it is transferred to disk.
   - The delta set in the R1's Capture cycle is moved to the R1's Transmit cycle, from which it begins transferring to the R2's Receive cycle.
   - A new delta set is created as the R1 Capture cycle, to collect host writes. The delta set is received on the R2 side.

   Subsequent host writes are collected into the next delta set.

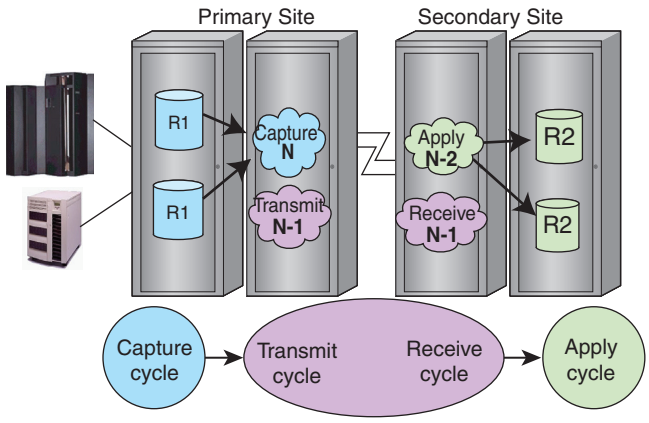


Figure 11 SRDF/A legacy mode

Mixed configurations

When one array in an SRDF configuration is running HYPERMAX OS, and one or more other arrays are running Enginuity 5876:

- SRDF/A single sessions (SSC) have only two cycles on the R1 side (legacy mode)
- SRDF/A multi-session consistency sessions (MSC) operate in legacy mode.

When a delta set is applied to the R2 target device, the R1 and R2 are in the *consistent* pair state. The R2 side is consistently 2 cycles behind the R1 site.

In the event of a failure at the R1 site or of the SRDF links, a partial delta set of data can be discarded, preserving consistency on the R2. The maximum data loss of for such failures is two SRDF/A cycles or less.

Multiple devices or device groups that require consistency can be grouped into *consistency groups*. Members of consistency groups cycle at the same time, to ensure consistency among the members, and if one member is interrupted, all other members suspend.

"SRDF consistency group operations" on page 155 provides more information.

HYPERMAX OS

If both arrays in the solution are running HYPERMAX OS, both SSC and MSC operate in multi-cycle mode. There can be 2 *or more* cycles on the R1, but only 2 cycles on the R2 side. Cycle switches are decoupled from committing delta sets from the R1 to the R2.

When the preset Minimum Cycle Time is reached, the R1 data collected during the capture cycle is added to the transmit queue and a new R1 capture cycle is started. There is no wait for the commit on the R2 side before starting a new capture cycle.

The transmit queue holds cycles waiting to be transmitted to the R2 side. Data in the transmit queue is committed to the R2 receive cycle when the current transmit cycle and apply cycle are empty.

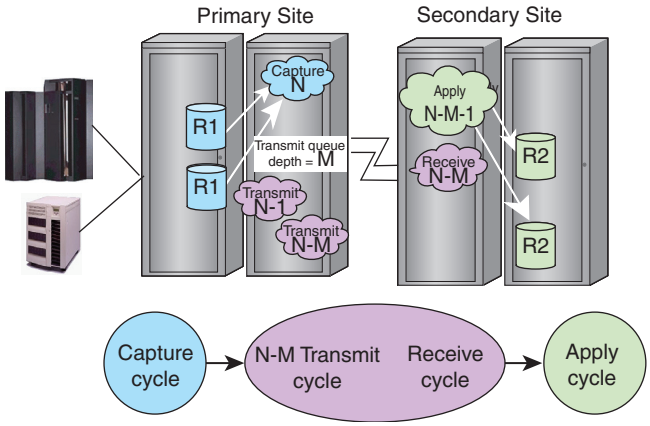


Figure 12 SRDF/A multi-cycle mode

Queuing allows smaller cycles of data to be buffered on the R1 side and smaller delta sets to be transferred to the R2 side.

The SRDF/A session can adjust to accommodate changes in the solution. If the SRDF link speed decreases or the apply rate on the R2 side decreases, more SRDF/A capture cycles can be added to the R1 side.

Data on the R2 side can be more than 2 cycles behind the R1.

In event of R1 failure or link failure, a partial delta set of data can be discarded, preserving consistency on the R2. The maximum data loss of for such failures can be more than two SRDF/A cycles.

EMC VMAX3 Family Product Guide provides a detailed description of SRDF/A multi-cycle mode.

SRDF/A write pacing, transmit idle, and Delta Set Extension

Unexpected write volume, transient link outages and other problems can result in the SRDF/A session being dropped. The following features help prevent session drop:

- Write pacing throttles host I/O to prevent cache overflow.

  You can enable/disable write pacing for a group or a device.

  "Managing SRDF/A write pacing" on page 141 provides more information.

- Transmit idle allows SRDF/A to wait a specified interval before dropping a session due to an SRDF link outage.

  Transmit idle allows the SRDF/A session to survive transient link outages.

  "Managing transmit idle" on page 141 provides more information.

- Delta Set Extension (DSE) pools page tracks to save pools when the array's write pending count exceeds the array's write pending limit (`-threshold`).

  DSE pools help SRDF/A manage unexpected spikes in I/O.

  "Managing Delta Set Extension" on page 133 provides more information.

SRDF/A benefits

- Provides remote mirroring at extended distances without adding response time to the R1 host. SRDF/A response times are equivalent to local non-SRDF devices.
- Supports efficient bandwidth usage.

  During the interval when SRDF collects host writes into a delta set, if the host writes the same track multiple times, SRDF sends only the latest change over the link.
- Supports all current SRDF topologies, including point-to-point and switched fabric.

  Supports all hosts and data emulation types supported by the array (FBA, CKD, AS 400).
- Provides dependent write consistency.

  Many applications (in particular, DBMS), use dependent write logic to ensure data integrity in the event of a failure. A dependent write is a write that is not issued by the application unless some prior I/O has completed. For example, when a DBMS updates a database:

  1. It first writes to the disk containing the log to indicate an update is coming, then
  2. It writes the data to the database dataset, and finally
  3. It writes again to the log volume to indicate that the database update is complete.

  These three writes (log, database, and log again) are related and each successive I/O is not issued until the prior I/O has successfully completed. If the writes are out of order, and an event such as a failure, or a creation of a point in time copy happens at that exact time, unrecoverable data loss may occur.

  SRDF/A ensures that dependent writes are in the same delta set or that later writes are in later delta sets.

SRDF /A features

- Supports multiple SRDF/A sessions per array.

  You can configure *all* your SRDF groups as SRDF/A-capable.
- Supports database consistency with SRDF/A-capable devices in composite groups using Multi-session Consistency (MSC).

  "SRDF consistency group operations" on page 155 provides more information.
- Maintains data consistency when mode is switched from asynchronous to synchronous.

  "Enabling/disabling consistency for SRDF/A devices" on page 128provides more information.
- Allows you to dynamically modify session priority, minimum cycle time, transmit idle and DSE pools.

HYPERMAX OS

For arrays running HYPERMAX OS, use the `symrdf set` command to dynamically change SRDF/A settings.

Enginuity 5773 - 5876

For arrays running Enginuity 5773-5876, use the `symconfigure` command to dynamically change SRDF/A settings.

- Allows you to dynamically add/remove device pairs from an active SRDF/A session without affecting the state of the session, or the reporting of SRDF pair states for devices that are not the target of the operation.

  "Adding/removing devices using consistency exempt" on page 129provides more information.

- Improves link utilization using software and hardware compression (Enginuity 5875 and higher).

Restrictions

- If either the R1 side or R2 side of an SRDF/A session is running HYPERMAX OS, Solutions Enabler 8.0.1 or higher is required to monitor MSC groups.
- All SRDF/A-capable devices running in asynchronous mode must be managed together in an SRDF/A session.
- For SRDF/A-capable devices enabled for consistency group protection, consistency must be disabled before attempting to change the mode from asynchronous.
- SRDF Automated Replication (SRDF/AR) control operations are currently **not** supported for SRDF/A-capable devices running in asynchronous mode.
- All SRDF/A sessions enabled within a consistency group operate in the same mode, multi-cycle or legacy. For example, if:
  - SRDF group 1 connects Site A and Site B, both running HYPERMAX OS, and
  - SRDF group 2 Site A running HYMPERMAX OS and Site C running 5876.
    - Group 1 can run in multi-cycle mode.
    - Group 2 must run in legacy mode.

  If both groups are in the same consistency group and are enabled together, then group 1 will transition from multi-cycle to legacy mode as a part of the enable.

- If there are tracks owed from the R2 to the R1, do not switch to asynchronous mode.

  Note: If tracks are owed to the R1 device, the -force option is required to make SRDF/A-capable devices in asynchronous mode Ready on the link.

TimeFinder snaps and clones

- TF/Snap and TF/Clone operations affect whether SRDF devices are allowed to be set in asynchronous mode.
- TF/Snap and TF/Clone pair states impact setting SRDF devices to asynchronous mode.

  "SRDF operations and TimeFinder sessions" on page 397 provides more information.

- Some Snap and Clone operations are not be allowed SRDF/A-capable devices operating in asynchronous mode.

  EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide provides more information.

Move operations

After a `movepair` or `half_movepair` action, the resulting SRDF mode for the moved device will be as follows:

- When moving a device *to* an SRDF group that is currently in asynchronous mode, the resulting SRDF mode for the device being moved will be asynchronous.
- When moving a device *from* an SRDF group in asynchronous mode, the resulting SRDF mode for the device being moved is synchronous.

Using BCVs to protect the R2 side

EMC recommends that you use TimeFinder BCVs at the remote site to mirror R2 devices. This practice preserves a consistent image of data before resynchronization operations.

R2 device BCVs can be consistently split off of the R2 without dropping the SRDF links or disrupting to the SRDF/A operational cycles.

R2 BCVs can be controlled from the R1-side or the R2-side host as long as the device groups have been defined on that host.

EMC Solutions Enabler TimeFinder Family CLI User Guide provides more information.

Immediate option

By default, the `failover`, `split`, and `suspend` operations cause SRDF to wait until the current cycle completes before dropping the session and making the devices Not Ready on the link. Completion time for these operations may be quite long.

Use the `-immediate` option in conjunction with `failover, split,` or `suspend` commands to immediately drop the SRDF/A session and make the devices Not Ready on the link.

The devices remain in asynchronous mode and pending tracks are converted to invalid tracks.

Use the `symrdf query -rdfa` command to display the number of tracks not committed to the R2 side and invalid tracks.

Restrictions

- The `-immediate` option applies only to devices participating in an active SRDF/A session.
- The `-immediate` option may result in remote invalid tracks on both the R1 and the R2 sides.
- The `-immediate` option does not compromise the consistency of data on the R2 side, but requires operator intervention to resolve any invalid tracks by using the correct `symrdf` command and pair state.
- If consistency is enabled on SRDF/A-capable devices, the `-force` option must used.

SRDF/Asynchronous operations

All SRDF/A operations (with the exception of consistency exempt, discussed later) must be performed on all devices in an SRDF group.

Thus, all devices in an SRDF group must be in the same SRDF device group. This is in contrast with SRDF/S, where operations can be performed on a subset of devices in an SRDF group.

Table 13 summarizes the operations described in this chapter.

Table 13 SRDF/A control operations

Control operation	Command	Description
Transition to asynchronous mode page 125	symrdf set mode async	Change the mode of the an SRDF group, composite group or device list to asynchronous mode.
Set SRDF group attributes page 126	symrdf set rdfa	Set the cycle time, session priority, and transmit idle for and SRDF/A group.
Check R1 devices for invalid tracks page 127	symrdf verify -noinvalids -consistent	Verify whether invalid tracks exist on both the R1 and R2 devices for a SRDF group, composite group or devices in a device list.
Enable/disable consistency for SRDF/A devices page 128	symrdf enable	Enable/disable consistency for a device group or devices in a device list.
Add/remove devices using the consistency exempt option page 129	symrdf createpair symrdf suspend symrdf movepair symrdf resume symrdf verify	Dynamically add and remove device pairs from an active SRDF/A session.
Display checkpoint complete status page 132	symrdf checkpoint	Display a checkpoint complete status when the data in the current cycle is committed to the R2 side.
Manage DSE page 133	symrdf set rdfa_dse symconfigure commit symcfg show	Set the SRDF/A DSE attributes for an SRDF group. Enginuity 5786 only: Add/remove/enable devices in DSE pools. Associate a DSE pool with and SRDF group. Monitor/display DSE pools.
Activate/deactivate DSE page 139	rdfa_dse_autostart symrdf activate/deactivate	Activate/deactivate SRDF/A DSE.
Enable transmit idle page 141	symrdf set rdfa -transmit_idle	Allow SRDF/A sessions to manage transient link outages without dropping.
Throttle host I/O page 143	symrdf set rdfa_pace symrdf -rdfa_pace activate symrdf -rdfa_pace deactivate symrdf -rdfa_wpace_exempt	Enable SRDF/A write pacing for groups or devices.
Display SRDF/A sessions page 148	symdg show symrdf -g *DgName* query -rdfa	Display SRDF/A sessions. Display SRDF/A groups.
Display SRDF devices page 148	symrdf list -rdfa	List SRDF/A capable devices.

Transitioning to asynchronous mode

You can create a new device group and specify the mode as asynchronous, or you can transition an existing SRDF device or group to asynchronous from another mode. The time it takes for devices or groups to transition from one mode to asynchronous mode varies depending on the original mode:

- From synchronous mode:

  If the devices are in a Synchronized state, the R2 devices already have a consistent copy.

  Enabling SRDF/A provides consistent data on the R2 immediately.

- From adaptive copy disk mode:

  Invalid tracks owed to the R2 are synchronized.

  Enabling SRDF/A provides consistent data on the R2 in two cycles.

- From adaptive copy write pending mode:

  Write pending slots are merged into the SRDF/A cycles.

  Enabling SRDF/A provides consistent data on the R2 two cycles after there are no more write pending slots.

symrdf set mode async command syntax

You can set the mode to asynchronous for a device group, composite group, or devices in a device file:

```
symrdf -g DgName set mode async
symrdf -cg CgName set mode async
symrdf -file Filename set mode async
```

Examples To set device group `prod` to asynchronous mode:

```
symrdf -g prod set mode async
```

To set composite group `Comp` to asynchronous mode:

```
symrdf -cg Comp set mode async
```

To set the devices listed in `Device.txt` to asynchronous mode:

```
symrdf -file device.txt set mode async
```

Note: This operation may not be allowed on TimeFinder/Snap and TimeFinder/Clone device pairs. "set mode async" on page 428, and page 429 provides more information.

Transitioning to synchronous mode

You can transition an SRDF/A device or device group to synchronous mode without losing consistency. Consistency on the R2 side is preserved.

The amount of time to complete the transition varies depending on whether the mode is legacy or multi-cycle:

- In legacy mode, the switch from asynchronous to synchronous requires two SRDF/A cycle switches to complete.

- In multi-cycle mode, the amount of time required includes the time to commit the current capture cycle and all cycles currently in the transmit queue to the R2 side.

Switching from asynchronous to synchronous is supported on Enginuity 5671 and higher.

symrdf set mode sync -consistent command syntax

You can set the mode to synchronous for a device group, storage group, or devices in a device file:

```
symrdf -g DgName -consistent set mode sync
symrdf -sg SgName -consistent set mode sync
symrdf -file Filename -consistent set mode sync
```

Examples

To switch modes from asynchronous to synchronous and maintain R2 data consistency in group `prod`:

```
symrdf -g prod -consistent set mode sync
```

To switch modes from asynchronous to synchronous and maintain R2 data consistency for devices listed in device file `devfile1`:

```
symrdf -f devfile1 -consistent set mode sync
```

Setting SRDF/A group cycle time, priority, and transmit idle

SRDF/A configuration parameters include array-wide parameters, and group level settings.

EMC Solutions Enabler Array Management CLI User Guide describes the steps to set the following SRDF/A array-wide parameters:

- SRDF/A cache usage - The percentage of write pending slots available to SRDF/A. Raising the value increases how much cache SRDF/A can use. Lowering the value reserves additional cache for non-SRDF/A cache usage.
- Maximum host throttle time - When the write pending limit is reached, delays writes from the host until a cache slot becomes free.

symrdf set rdfa command syntax

To set the SRDF/A group-level attributes on an SRDF group:

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
     [-noprompt] [-i Interval] [-c Count]

.............

set rdfa
   [-cycle_time 1 - 60]
   [-priority 1 - 64]
   [-transmit_idle {on|off}]
   [-both_sides]
```

Where:

- `-cycle_time` (`-cyc`) is the minimum time to wait before attempting an SRDF/A cycle switch.

  Valid values are 1 through 60 seconds.

  Default value for Enginuity 5874 and higher is 15.

Default value for Enginuity lower than 5874 is 30.

See "Enabling and disabling SRDF consistency protection" on page 158 for details about how minimum cycle times are handled when you enable consistency protection.

- `-priority` (`-pri`) determines which SRDF/A sessions are dropped if the cache becomes full.

  Valid values are 1 (highest priority, last to be dropped) through 64 (lowest priority).

  The default value is 33.

- `-transmit_idle` (`-tra`) allows the SRDF/A session to wait (not drop) when the link cannot transmit data.

  Valid state values are on and off.

  Default is on.

- `-both_sides` applies the SRDF/A attributes to both the source and target sides of an SRDF/A session. If `-both_sides` is not specified, attributes are only applied to the source side.

Examples

To set the minimum cycle time for both sides of SRDF/A group 160:

```
symrdf -sid 134 -rdfg 160 set rdfa -cycle_time 32 -both_sides
```

To set the session priority for both sides of SRDF/A group 160:

```
symrdf -sid 134 -rdfg 160 set rdfa -priority 55 -both_sides
```

To set the cycle time and session priority for only the source side of SRDF/A group 12:

```
symrdf -sid 134 -rdfg 12 set rdfa -cycle_time 32 -priority 20
```

```
An RDF Set 'Attributes' operation execution is in progress for RDF group 12.
Please wait...

    SRDF/A Set Min Cycle Time(1134,012)........................Started.
    SRDF/A Set Min Cycle Time (1134,012)......................Done.
    SRDF/A Set Priority (1134,012)............................Started.
    SRDF/A Set Priority (1134,012)............................,,,,Done.
The RDF Set 'Attributes' operation successfully executed for RDF group 12.
```

Checking for R1 invalid tracks

Under normal operations, the `symrdf verify -consistent` command verifies that SRDF device pairs are in the R2 Consistent pair state. No invalid tracks are owed to the R2 side from its R1 side.

When an SRDF pair is in the Split state and the host writes to its R2 device, invalid tracks are owed to its R1 device.

When the pair is restored, the pair is still in the Consistent state because no invalid tracks are owed to the R2 device. SRDF does not recognize invalid tracks owed from R2 to R1.

Use the `-noinvalids` and `-consistent` options to perform an addition check to verify whether invalid tracks exist on both the R1 and R2 devices.

symrdf verify command syntax

You can issue the `symrdf verify` command with `-noinvalids` and `-consistent` options, to device groups, composite groups, storage groups, and device files.

```
symrdf verify -g DgName -consistent -noinv
symrdf verify -cg CgName -consistent -noinv
symrdf verify -sg SgName -consistent -noinv
symrdf verify -file DeviceFile -consistent -noinv
```

Example To monitor the clearing of invalid tracks every 60 seconds for the device group `dg1`:

```
symrdf verify -g dg1 -consistent -noinv -i 60

None of the devices in the group 'dg1' are in 'Consistent with no
   invalid tracks' state.

Not all devices in the group 'dg1' are in 'Consistent with no invalid
   tracks' state.

All devices in the group 'dg1' are in 'Consistent with no invalid
   tracks' state.
```

Enabling/disabling consistency for SRDF/A devices

The consistency feature ensures the dependent-write consistency of the data distributed across multiple R1 devices. The R1 and R2 devices can be distributed across multiple primary and secondary arrays.

Consistency groups are groups of SRDF devices enabled for database consistency. SRDF devices that belong to the same consistency group act in unison to preserve dependent-write consistency of a database distributed across multiple devices within the consistency group.

The consistency group ensures that remote mirroring is suspended for all SRDF devices in a consistency group as soon as one SRDF device in the group fails to send data across the SRDF links.

- Use the `enable` argument to enable consistency protection for devices in SRDF/Asynchronous mode by device group or device list.

  When consistency is enabled, and data cannot be copied from the R1 to the R2, all devices in the group will be made not ready on the links.

- Use the `disable` argument to disable consistency protection for devices in SRDF/Asynchronous mode by device group or device list.

  When consistency is disabled, and data cannot be copied from the R1 to the R2, only the devices in the group that are experiencing problems will be made not ready on the links. The device state for any remaining devices in the group will remain the same.

"SRDF consistency group operations" on page 155 describes how to enable consistency protection for SRDF/A pairs in a composite group.

symrdf enable command syntax

You can enable consistency for SRDF/A device pairs in a device group, storage group, or devices in a device file:

```
symrdf -g DgName -sid SID -rdfg RdfgNum enable
```

```
symrdf -sg SgName -sid SID -rdfg RdfgNum enable
symrdf -file Filename -sid SID -rdfg RdfgNum enable
```

IMPORTANT

For concurrent SRDF configurations, you must enable consistency for each R2 mirror separately.

To use the `-file Filename` option:

- All device pairs in that SRDF group must be in the device file.
- If the device file includes concurrent devices, only the R2 side specified by the `-sid SID -rdfg` arguments is enabled.

  The device group on the second R2 side is not enabled.

To use the `-g DgName` option:

- All device pairs in that SRDF group must be in the device group.
- If `the device group i`ncludes concurrent devices, only the R2 side specified by the `-sid SID -rdfg` arguments is enabled.

Restrictions

Because you must enable consistency for each R2 mirror separately in a concurrent relationship, you cannot use the `-rdfg all` option.

Examples

To enable consistency protection for SRDF/A pairs in device group `prod`:

```
symrdf -g prod enable
```

To enable consistency protection for SRDF/A pairs listed in device file `devfile1`:

```
symrdf -file devfile1 -sid 123 -rdfg 10 enable
```

To disable consistency protection for SRDF/A pairs in device group `prod`:

```
symrdf -g prod disable
```

To disable consistency protection for SRDF/A pairs listed in device file `devfile1`:

```
symrdf -file devfile1 -sid -rdfg 10 disable
```

To enable consistency for devices in device file `file1`:

```
symrdf -f file1 -sid 123 -rdfg 55 enable
```

To enable consistency for R2 devices in a concurrent configuration (SRDF group 56 and SRDF group 57) of `devgroup2`:

```
symrdf -g devgroup2 -rdfg 56 enable
symrdf -g devgroup2 -rdfg 57 enable
```

Adding/removing devices using consistency exempt

Note: The consistency exempt option is available with Enginuity 5773.150 and higher.

The consistency exempt option (`-cons_exempt`) allows you to dynamically add and remove device pairs from an active SRDF/A session without affecting:

- the state of the session, or
- the reporting of SRDF pair states for devices that are not the target of the operation

When enabled, the consistency exempt option places devices into a consistency exempt state. Exempt devices are excluded from the group's consistency check.

After the operation is complete, the consistency exempt state is automatically terminated. Specifically, consistency is terminated when:

- The target devices are resumed and fully synchronized and
- Two full cycle switches have occurred, or

  The devices are removed from the group.

You can use the `-cons_exempt` option with the following commands:

- `createpair`

  The SRDF pairs become consistency exempt in the SRDF group in which they are created.

- `movepair`, `half_movepair`

  The SRDF pairs become consistency exempt in the target SRDF group into which they are moved.

- `suspend`

  Device pairs become consistency exempt in their current SRDF group. Device pairs moved from one group to another can be suspended with consistency exempt without effecting other devices in their group.

When devices are suspended and consistency exempt (within an active SRDF/A session) they can be controlled apart from other devices in the session. This is useful for `resume`, `establish`, `deletepair`, `half_deletepair`, `movepair`, and `half_movepair` operations.

Restrictions

- The consistency exempt option cannot be used:
  - For devices that are part of an SRDF/Star configuration.
  - For an SRDF/A session that is in the Transmit Idle state.
- If the device is an R2 device and the SRDF/A session is active, the `half_movepair` and `half_deletepair` commands are blocked.
- If the session is deactivated before the consistency exempt state is cleared, when re-activated, the device remains in the consistency exempt state until the device has no invalid tracks that need to be synchronized.
- A `movepair` operation of an SRDF pair to another SRDF group with an active SRDF/A session is only allowed when the SRDF pair state is suspended and can be blocked if in the failed over or split pair state.
- The `createpair` and `movepair` operations are allowed without the `-cons_exempt` option if the new SRDF group is operating in the asynchronous mode but the SRDF/A session is not active.

Adding device pairs to an active SRDF/A session

Procedure The following example uses device file “Myfile” to add device pairs to an active SRDF/A session:

1. Use the `createpair -establish` command to create the new device pairs, add them to a temporary SRDF group (10), and synchronize:

   `symrdf createpair -file Myfile -sid 1234 -rdfg 10 -type RDF1 -establish`

2. Use the `verify -synchronized` command to monitor synchronization:

   `symrdf verify -file MyFile -sid 1234 -rdfg 10 -synchronized`

   When the device pairs are synchronized:

3. Use the `suspend` command to suspend the device pairs in th temporary group so they can be moved to the active SRDF/A group:

   `symrdf suspend -file MyFile -sid 1234 -rdfg 10`

   Note: The temporary group is synchronous, thus the consistency exempt option is not allowed.

4. Use the `movepair` command with the `-cons_exempt` option to move the device pairs from the temporary SRDF group to the active SRDF/A group:

   `symrdf movepair -file MyFile -sid 1234 -rdfg 10 -new_rdfg 20 -cons_exempt`

5. Use the `resume` command to resume the device pairs:

   `symrdf resume -file MyFile -sid 1234 -rdfg 20`

6. Use the `verify -consistent -noinvalids` command to display when the device pairs become consistent and there are no invalid tracks on the R1 and R2 sides:

   `symrdf verify -file MyFile -sid 1234 -rdfg 20 -consistent -noinvalids`

   ⚠CAUTION

   Do not enable host access to the R1 side until the pair state for the devices reaches Consistent.

Removing device pairs from an active SRDF/A session

Procedure The following example uses device file “Myfile” to remove device pairs from an active SRDF/A session:

1. Use the `suspend` command with the `-cons_exempt` option to suspend the device pairs to be removed:

   `symrdf suspend -file MyFile -sid 1234 -rdfg 20 -cons_exempt`

2. Use the `movepair` command to move the device pairs from the current SRDF group to another SRDF group:

   `symrdf movepair -file MyFile -sid 1234 -rdfg 20 -new_rdfg 30`

3. Use the `resume` command to resume the devices in their new group:

   `symrdf resume -file MyFile -sid 1234 -rdfg 30`

4. Use the `verify -synchronized` command to monitor synchronization:

```
symrdf verify -file MyFile -sid 1234 -rdfg 30 -synchronized
```

⚠CAUTION

Do not enable host access to the R1 side until the pair state for the devices reaches Consistent.

Displaying checkpoint complete status

Use the `checkpoint` argument to display a checkpoint complete status when the data in the current cycle is committed to the R2 side.

The target devices must be in an active SRDF/A session.

symrdf checkpoint command syntax

You can issue the `checkpoint` operation on a device group, composite group, storage group, and device file:

```
symrdf -g DgName [-i Interval] [-c Count] [-rdfg GrpNum]
     [ -hop2 | -bcv [-hop2] | -all | -rbcv | -brbcv] checkpoint

symrdf -cg CgName [-i Interval] [-c Count]
     [ -hop2 ] [-rdfg SymmID:GrpNum | name:RdfGroupName] checkpoint

symrdf -sg SgName -sid SymmID -rdfg GrpNum
      [-i Interval] [-c Count] checkpoint

symrdf -file Filename -sid SymmID -rdfg GrpNum
      [-i Interval] [-c Count] checkpoint
```

Where:

- `-c Count` specifies the number of times (Count) to repeat the operation before exiting.
- `-i Interval` specifies the number of seconds to wait between successive iterations of the operation.

  Default: 10 seconds.

  Minimum interval: 5 seconds.

If `-c Count` is not specified and `-i Interval` is specified, the operation repeats continuously at the specified interval.

If `-c Count` is specified and `-i Interval` is not specified, the operation repeats the specified number of iterations using the default interval.

Restrictions

- For All specified devices must be in the same SRDF/A session.

Examples

To confirm R2 data copy for device group `prod`:

```
symrdf -g prod checkpoint
```

To confirm the R2 data copy for devices in device group `Test` in RA group 7 on the second hop of a cascaded SRDF configuration:

```
symrdf -g Test -rdfg 7 -hop2 checkpoint
```

Managing Delta Set Extension

When many SRDF/A groups run on the same array, complex I/O profiles and link availability/bandwidth issues complicate the task of calculating cache requirements.

SRDF/A Delta Set Extension (DSE) extends the cache space available for SRDF/A session cycles by off loading cycle data from cache to preconfigured pool storage. DSE helps SRDF/A to ride through larger and longer throughput imbalances than cache-based buffering alone.

DSE is enabled by default on arrays running HYPERMAX OS, and disabled by default on arrays running Enginuity 5773 - 5876.

Note: DSE is not designed to solve permanent or persistent problems such as unbalanced/insufficient cache, host writes that consistently overrun cache, and long link outages.

When the SRDF/A session is activated, DSE is activated (on the R1 and R2 sides) if the autostart for DSE is set to enabled on both the R1 and the R2 sides. Autostart for DSE can be enabled/disabled, but the change does not take effect until the SRDF/A session is dropped and re-activated. By default, autostart for DSE is enabled regardless of whether the side is the R1 or R2 side.

DSE starts paging SRDF/A tracks to the DSE pool when the array write pending count crosses the DSE threshold (-threshold attribute). The default threshold is 50 percent of the System Write Pending Limit. After a cycle switch, Enginuity reads tracks from the DSE pool back into the array cache so that they can be transferred to the R2.

Enginuity 5733 -5876

Arrays running Enginuity 5773 - 5876, can share SRDF/A DSE pools among multiple SRDF/A groups. A single SRDF/A group can have up to 4 DSE pools associated with it (one for each device emulation type).

"Managing DSE pools (Enginuity 5773 - 5876)" on page 135 describes best practices, restrictions and procedures to manage DSE pools.

HYPERMAX OS

Arrays running HYPERMAX OS come preconfigured with one or more Storage Resource Pools (SRPs) containing all the storage available to the array. SRDF/A DSE allocations are made against one SRP per array designated as the SRP for DSE.

The SRP designated for DSE supports the DSE allocations for all SRDF/A sessions on the array.

The default SRP for DSE is the default SRP for FBA devices.

You can change which SRP is associated with DSE, and you can change the capacity of the SRP associated with DSE.

- "Managing the DSE SRP capacity (HYPERMAX OS)" on page 134 describes the procedures to manage the capacity of the DSE SRP.
- *EMC Solutions Enabler Array Management CLI User Guide* describes the steps to modify which SRP is associated with DSE.

Managing the DSE SRP capacity (HYPERMAX OS)

This section describes the steps to modify the capacity of the DSE SRP for arrays running HYPERMAX OS, including:

- "Restrictions" on page 134
- "About the DSE SRP" on page 134
- "Modify the DSE SRP capacity" on page 134

Restrictions

- CFGSYM access rights and Storage Admin authorization rights are required to run the symconfigure set command.
- If DSE requests for allocations exceed the maximum capacity of the DSE SRP, the SRDF/A session may drop.
- HYPERMAX OS does not support user defined DSE pools, and the following symrdf set commands are not supported:
  - symrdf set rdf_dse –fba_pool
  - symrdf set rdf_dse –ckd3390_pool
  - symrdf set rdf_dse –ckd3380_pool
  - symrdf set rdf_dse –as400_pool

About the DSE SRP

The default SRP associated with DSE is configured prior to installation. You can create another SRP for use with DSE, but only one SRP per array can be associated with DSE. All SRDF/A sessions on the array use the one SRP designated for use with DSE.

- If you enable SRDF/A DSE (rdfa_dse attribute) on another SRP, that SRP becomes the SRP for all DSE allocations.

  The SRP that was previously designated to support DSE is automatically modified not to support DSE (its rdfa_dse attribute is set to disabled).
- If you disable the rdfa_dse attribute on the DSE SRP without designating another SRP to support DSE, the default SRP for FBA emulation automatically becomes the DSE SRP.

Modify the DSE SRP capacity

Use the `symconfigure set symmetrix dse_max_cap` command to modify the capacity of the DSE SRP.

symconfigure set symmetrix dse_max_cap command syntax

```
symconfigure -sid SID commit -cmd "set symmetrix dse_max_cap =
MaxCap;"
```

Where:

- *`MaxCap`* specifies the maximum capacity in the array's DSE SRP. Valid values are:
  - 1 - 100000 - Specifies the maximum number of GB in the specified SRP that can be used by DSE.

- NoLimit - Specifies that DSE can use the entire capacity of the specified SRP.

Examples To set the maximum DSE capacity on SID 230 to a value of 100 GB:

```
symconfigure -sid 230 commit -cmd "set symmetrix dse_max_cap = 100;"

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 symmetrix dse_max_cap=100;
    }

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

The configuration change session has successfully completed.
```

To set the maximum DSE capacity on SID 230 to unlimited:

```
symconfigure -sid 230 commit -cmd "set symmetrix dse_max_cap = nolimit;"

Execute a symconfigure operation for symmetrix '000197100230' (y/[n]) ? y
.
.
.
The configuration change session has successfully completed.
```

Managing DSE pools (Enginuity 5773 - 5876)

This section describes DSE pool management for arrays running Enginuity 5773 - 5876, including:

- "DSE pool best practices"
- "Setting SRDF/A DSE attributes for an SRDF group"
- "Clearing existing DSE pool names"
- "Adding devices to an SRDF/A DSE pool"
- "Removing devices from an SRDF/A DSE pool"
- "Enabling/disabling devices in an SRDF/A DSE pool"
- "Associating an SRDF group with a DSE pool"
- "Displaying/monitoring SRDF/A DSE pool usage"

IMPORTANT

The procedures in this section are not applicable to VMAX 3 arrays running HYPERMAX OS. For the procedures applicable to VMAX 3 arrays, refer to "Managing the DSE SRP capacity (HYPERMAX OS)" on page 134

Restrictions

- A DSE pool cannot have the same name as a Snap pool on the same array.
- Each DSE pool can only contain one type of device emulation: FBA, CKD3390, CKD3380, or AS400.

- Each SRDF group can have at most one pool of each emulation.

DSE pool best practices

- Configure DSE pools on both R1 and R2 arrays.
- Plan for peak workloads.
- Spread the DSE pool devices across as many disks as possible.
- Ensure that sufficient DA and RA CPU resources are available for the DSE task.
- To simplify management and make the most efficient use of resources, use as small a number of DSE pools as possible.
- Configure DSE pools and enable DSE on the primary and on the secondary array. When TimeFinder/Snap sessions are used to replicate either R1 or R2 devices, create two separate preconfigured storage pools: DSE and Snap pools.
- Configure a separate DSE pool for each device emulation type (FBA, IBMi, CKD3380 or CKD3390). You can create multiple DSE pools for different SRDF/A groups.

Best Practices for EMC® SRDF®/A Delta Set Extension Technical Note provides more information.

Setting SRDF/A DSE attributes for an SRDF group

Use the `set rdfa_dse` command to set the SRDF/A DSE attributes for an SRDF group.

Note: The remote array must be reachable to complete this task.

symrdf set rdfa_dse command syntax

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
     [-noprompt] [-i Interval] [-c Count]

.............

set rdfa_dse
   [-autostart {on | off}]
   [-threshold 20 - 100]
   [-fba_pool PoolName]
   [-ckd3390_pool PoolName]
   [-ckd3380_pool PoolName]
   [-as400_pool PoolName>]
   [-both_sides]
```

Where:

- `-autostart` (`-aut`) specifies whether SRDF/A DSE is automatically enabled or disabled when an SRDF/A session is activated for an SRDF group.

  Valid values are on or off.

  Default is off.

- `-threshold` (`-thr`) specifies the percentage of the array's write pending limit. If cache usage of all active SRDF/A groups in the array exceeds this limit, data tracks for this SRDF group start to spill over to disks.

  Valid values 20 - 100.

Default is 50.

- `-fba_pool` (`-fba`) *`PoolName`* associates the pool *`PoolName`* containing SAVE devices with FBA emulation with the specified SRDF group.

  If the argument *`PoolName`* is not specified, the currently associated FBA pool is removed from the group.

- `-ckd3380_pool` (`-ckd3380`) *`PoolName`* associates the pool *`PoolName`* containing SAVE devices with CKD 3380 emulation with the specified SRDF group.

  If the argument *`PoolName`* is not specified, the currently associated CKD 3380 pool is removed from the group.

- `-ckd3390_pool` (`-ckd3380`) *`PoolName`* associates the pool *`PoolName`* containing SAVE devices with CKD 3390 emulation with the specified SRDF group.

  If the argument *`PoolName`* is not specified, the currently associated CKD 3390 pool is removed from the group.

- `-as400_pool` (`-as400`) *`PoolName`* associates the pool *`PoolName`* containing SAVE devices with an AS400 emulation with the specified SRDF group.

  If the argument *`PoolName`* is not specified, the currently associated AS400 pool is removed from the SRDF group.

- `-both_sides` sets the SRDF/A DSE attributes on both the source and target sides of an SRDF/A session.

  If `-both_sides` is not specified, attributes are only applied to the source side.

For arrays running Enginuity 5876, you can also use the `symconfigure` command to set these SRDF/A DSE attributes. See the *EMC Solutions Enabler Array Management CLI User Guide.*

Clearing existing DSE pool names

Use the `-_pool` commands with no *`PoolName`* argument to remove the association between the specified SRDF group and DSE pools.

Example To clear the DSE pool names for all 4 emulation types:

```
symrdf -sid 432 -rdfg 75 set rdfa_dse -fba_pool -ckd3390_pool -ckd3380_pool
   -as400_pool

An RDF Set 'Attributes' operation execution is in progress for
RDF group 75. Please wait...

    SRDF/A Set FBA Pool (0432,075)....................................Started.
    SRDF/A Set FBA Pool (0432,075)....................................Done.
    SRDF/A Set CKD3380 Pool (0432,075)................................Started.
    SRDF/A Set CKD3380 Pool (0432,075)................................Done.
    SRDF/A Set CKD3390 Pool (0432,075)................................Started.
    SRDF/A Set CKD3390 Pool (0432,075)................................Done.
    SRDF/A Set AS400 Pool (0432,075)..................................Started.
    SRDF/A Set AS400 Pool (0432,075)..................................Done.

The RDF "Attributes'' operation successfully executed for RDF group 75.
```

Adding devices to an SRDF/A DSE pool

Devices can be added to a DSE pool if they are:

- Disabled
- Inactive
- Do not belong to another pool

Command syntax

To add and enable SAVE devices to a DSE pool:

```
add dev SymDevName [:SymDevName] to pool PoolName
type = <snap | rdfa_dse>
[, member_state = <ENABLE | DISABLE> ];
```

Example

```
add dev 018B:018C to pool finance,
type = rdfa_dse,
member_state=ENABLE;
```

Removing devices from an SRDF/A DSE pool

You can remove SAVE devices from an SRDF/A DSE pool, only if the devices are disabled and drained.

When you remove a device from a pool, it becomes available for use by other SAVE device pools.

The last device cannot be removed from an SRDF/A DSE pool if the pool is associated with an SRDF group.

Command syntax

```
remove dev SymDevName[:SymDevName]
from pool PoolName,
type = <snap | rdfa_dse>;
```

Example

```
remove dev 018B from pool finance, type = rdfa_dse;
```

Enabling/disabling devices in an SRDF/A DSE pool

Devices in a DSE pool do not all have to be in the same state (enabled or disabled):

- If all the devices in a pool are disabled, the pool is disabled.
- If at least one device in a pool is enabled, the pool is enabled.

To enable or disable a range of devices, all the devices must be in the same pool.

You cannot disable all the devices in an SRDF/A DSE pool if the pool is currently associated with an SRDF group and SRDF/A DSE is active for the group.

Command syntax

```
enable dev SymDevName[:SymDevName] in pool PoolName,
type = <snap | rdfa_dse>;
```

Example

```
enable dev 018C in pool finance,
type = rdfa_dse
```

Associating an SRDF group with a DSE pool

SRDF/A DSE pools are created and managed in command files that are executed using the `symconfigure` command.

Procedure To set the SRDF/A DSE threshold, associate an SRDF group with a pool, and activate DSE:

1. Use the `symcfg list -sid SID -pools -rdfa_dse` command to list the configured DSE pools.
2. Create a text file containing the commands to set attributes for an SRDF group.

   The first command in the file must be to set the threshold.

   To set the threshold, associate with DSE pool `r1pool`, and specify FBA emulation, and enable autostart for SRDF group 7:

   ```
   set rdf group 7 rdfa_dse_threshold=20;
   set rdf group 7 rdfa_dse_pool=r1pool, emulation=fba;
   set rdf group 7 rdfa_dse_autostart=enable;
   ```

3. Use the `symconfigure commit` command to perform the operation:

   ```
   symconfigure commit -sid 12 -file setup_dse.cmd
   ```

EMC Solutions Enabler Array Management CLI User Guide provides more information about the `symconfigure` command.

Displaying/monitoring SRDF/A DSE pool usage

Use the `symcfg show` command to display the pool utilization for a specified SRDF/A DSE pool.

Command syntax

```
symcfg show [-sid SymmID] -pool PoolName -rdfa_dse
```

Example To display the utilization for DSE pool BC_DSE:

```
symcfg show -sid 03 -pool BC_DSE -rdfa_dse
```

```
Symmetrix ID: 000194901103

Symmetrix ID                    : 000194901103
Pool Name                       : BC_DSE
Pool Type                       : Rdfa DSE
Disk Location                   : Internal
Technology                      : FC
Dev Emulation                   : FBA
Dev Configuration               : 2-Way Mir
Pool State                      : Enabled
# of Devices in Pool            :   4
# of Enabled Devices in Pool    :   4
# of Related RDF Groups         :   1

Enabled Devices(4):
  {
   ----------------------------------------------------
   Sym          Usable        Free        Used Full Device
   Dev          Tracks      Tracks      Tracks  (%) State
   ----------------------------------------------------
   0228          17250        8682        8568   49 Enabled
   0229          17250       17062         188    1 Enabled
   022A          17250        8413        8837   51 Enabled
   022B          17250        8579        8671   50 Enabled
            ---------- ---------- ---------- ----
   Tracks        69000       42736       26264   38
```

Activating/deactivating SRDF/A DSE

There are several methods to activate SRDF/A DSE:

- Set the SRDF/A group parameter `rdfa_dse_autostart` to ENABLE.

SRDF/A DSE becomes active when the SRDF/A session is activated.

- Modify the SRDF/A DSE status for a device group, composite group, or file when the SRDF link status is Read Write,

  This activates or deactivates SRDF/A DSE for groups on both the R1 and R2 sides.

  Note: The SRDF links must be in asynchronous mode and SRDF/A must be active for activate or deactivate actions to succeed.

  Use the following commands to modify the device group, composite group, or file:

  ```
  symrdf [-g DgName | -cg CgName | -f FileName]
  activate | deactivate -rdfa_dse
  ```

- Modify the SRDF/A DSE status using RA group operations when the SRDF link status is Read Write.

  Use the following commands to modify the group:

  ```
  symrdf -sid SymmID -rdfg GrpNum [-v][-noprompt] [-i Interval] [-c
  Count]

  activate   -rdfa_dse  [-both_sides]
  deactivate  -rdfa_dse [-both_sides]
  ```

  The `-both_sides` option activates/deactivates SRDF/A DSE for groups on both the source and target sides. Otherwise, the activate/deactivate is only performed on the source side.

- Set the group mode to `sync` or `acp` when SRDF/A DSE is active for an SRDF group.

  This method does not require deactivating SRDF/A DSE.

  Deactivating SRDF/A in a group automatically deactivates SRDF/A DSE for that group.

Requirements to activate SRDF/A DSE with Dynamic Cache Partitioning

Requirements for activating SRDF/A DSE with dynamic cache partitioning include:

- All devices in the SRDF/A session must be in the same DCP.
- The `rdfa_dse_threshold` must be set, and must be lower than the `rdfa_cache_percentage` setting.
- The SRDF group must have at least one associated DSE pool with SAVE devices enabled.

Use the following syntax to activate SRDF/A DSE when dynamic cache partitioning is enabled:

```
symrdf type activate -rdfa_dse
```

Valid values for *type* are `-dg`, `-cg`, `-file`, or `-rdfg`.

Note: After activation, R1 and R2 cache usage is reported as a percent of DCP Write Pending Limit.

Managing transmit idle

Transmit idle allows an SRDF/A session to manage transient link outages without dropping. If transmit idle is not enabled, the SRDF/A session drops when the link cannot transmit data.

If transmit idle is enabled, a link failure starts the link limbo timer. If the link status is still Not Ready after the link limbo time expires, devices are marked Not Ready to the link. When the link recovers, the SRDF pair state returns to Consistent or SyncInProg.

See also "Link limbo" on page 87 and "Setting SRDF group attributes" on page 91.

Restrictions

When the SRDF pair is in the Transmit Idle state, only the following operations are allowed from the R1 side:

- rw_enable -r1
- write_disable -r1
- ready -r1
- not_ready -r1
- suspend -immediate

When the SRDF pair is in the Transmit Idle state, only the following operations are allowed from the R2 side:

- suspend -immediate
- failover -immediate

If at the beginning of a control action, all SRDF/A groups are not in the Transmit Idle state, the action fails if one of the groups enters the Transmit Idle state during processing.

symrdf set_rdfa command syntax

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
[-noprompt] [-i Interval] [-c Count]
.............

   set rdfa
      [-transmit_idle {on | off}]
      [-both_sides]
```

Example To enable transmit idle on both sides for SRDF/A group 12:

```
symrdf -sid 134 -rdfg 12 set rdfa -transmit_idle on -both_sides
```

Managing SRDF/A write pacing

SRDF/A write pacing extends the availability of SRDF/A by preventing conditions that result in cache overflow on both the R1 and R2 sides. Write pacing balances cache utilization by extending the host write I/O response time to prevent SRDF/A operational interruptions.

There are two types of write pacing:

- **Group-level pacing** is dynamically enabled for the entire SRDF/A group when slowdowns in host I/O rates, transmit cycle rates, or apply cycle rates occur.

  SRDF/A group-level write pacing monitors and responds to:

- Spikes in the host write I/O rates
- Slowdowns in data transmittal between R1 and R2
- R2 restore rates.

Group-level pacing controls the amount of cache used by SRDF/A. This prevents cache overflow on both the R1 and R2 sides, and helps the SRDF/A session to stay up and running.

Group-level pacing requires Enginuity 5874 or greater.

HYPERMAX OS

HYPERMAX OS introduces enhanced group-level pacing. Enhanced group-level pacing paces host I/Os to the DSE transfer rate for an SRDF/A session.

When DSE is activated for an SRDF/A session, host-issued write I/Os are throttled so their rate does not exceed the rate at which DSE can offload the SRDF/A session's cycle data.

Enhanced group-level pacing requires HYPERMAX OS on the R1 side. The R2 side can be running either HYPERMAX OS or Enginuity 5876.

Enhanced group-level pacing responds only to the spillover rate on the R1 side. It is not affected by spillover on the R2 side.

Enginuity 5773-5876

- **Device-level pacing** is for SRDF/A solutions in which the SRDF/A R2 devices participate in TimeFinder copy sessions.

  Note: device-level pacing is not supported in HYPERMAX OS.

  SRDF/A device-level write pacing addresses conditions that lead to cache overflow specifically due to TimeFinder/Snap and TimeFinder/Clone sessions on an R2 device running in asynchronous mode.

  Device-level write pacing requires Enginuity version 5875.135.91 or higher on both arrays.

You can enable/disable either or both write pacing options. Both write pacing options are compatible with each other and with other SRDF/A features including tunable cache utilization, Reserve Capacity, and MSC.

Enginuity version 5876.82.57 or higher includes a global write pacing statistics report.

You can activate and control group-level and device-level write pacing individually or simultaneously at the group, device group, composite group, or file level on the R1 side.

Each has an autostart capability that automatically activates write pacing whenever an SRDF/A session becomes active. If an SRDF group has both group-level and device-level pacing configured to autostart, both are activated when the SRDF/A session becomes active.

For information on using group-level and device-level write pacing, see "Write pacing operations" on page 143.

Requirements

- Group-level pacing is supported on Enginuity 5874.207.166 and higher.
  - The group-level exemption capability requires that the R1 side be running Enginuity 5875 and higher.

- Enhanced group-level write pacing requires that the R1 and R2 arrays be running Enginuity 5876 and higher.
- Device-level write pacing is supported on Enginuity 5875 and higher.
- The `activate` argument requires that the SRDF/A session be active and contain at least one participating device.

  This requirements does not apply to the autostart capability.

Write pacing operations

SRDF/A write pacing bases some of its actions on the following:

- R1 side cache usage
- Transfer rate of data from transmit delta set to receive delta set
- Restore rate on the R2 side

Starting with Enginuity 5876 SRDF/A group-level write pacing can respond to the following conditions:

- The write-pending level on an R2 device in an active SRDF/A session reaches the device's write-pending limit.
- The restore (apply) cycle time on the R2 side is longer than the capture cycle time.

The enhanced group-level write pacing feature can effectively pace host write I/Os in the following operational scenarios:

- Slower restore (apply) cycle times on specific R2 devices that are managed by slower-speed physical drives.
- FAST operations that lead to an imbalance in SRDF/A operations between the R1 and R2 sites.
- Sparing operations that lead to R2-side DAs becoming slower in overall restore operations.
- Production I/Os to the R2 side that lead to DAs and/or RAs becoming slower in restore operations.
- Restore delays during the pre-copy phase of TimeFinder/Clone sessions before activation.

The configuration and management of group-level write pacing are unaffected by this enhancement.

Write pacing considerations

Group level write pacing

- Only the group-level pacing values configured for the SRDF group on the R1 side of the SRDF/A session are used.
- In a cascaded SRDF environment:
  - With Enginuity 5876 and lower, group-level write pacing is only supported on the R1->R21 hop of the relationship.

- With Enginuity 5876 Q4 2012 SR and higher, group-level write pacing is supported on both the R1->R21 and R21->R2 hops of the relationship.

- In a concurrent SRDF/A environment, group-level pacing is supported on both mirrors of the concurrent R1. In this case, write pacing calculations are performed independently for the two SRDF/A sessions, and the host write I/Os sessions are subject to the greater of the two calculated delays.

Device-level write pacing

- Only the device-level pacing values configured for the SRDF group on the R1 side of the SRDF/A session are used.
- In a cascaded SRDF environment:
  - With Enginuity 5876 and lower, device-level write pacing is only supported on the R1->R21 hop of the relationship.
  - With Enginuity 5876 Q4 2012 SR and higher, device-level write pacing is supported on both the R1->R21 and R21->R2 hops of the relationship.
- There is no exemption from device-level pacing as there is for group-level pacing, and the R1 group-level exempt state does not affect device-level pacing.
- In a concurrent SRDF/A environment, device-level pacing is supported on both mirrors of the concurrent R1. In this case, write pacing calculations are performed independently for the two SRDF/A sessions, and the host write I/Os sessions are subject to the greater of the two calculated delays.
- If both group-level pacing and device-level pacing are active for an SRDF/A session, the group-level and device-level delays are calculated independently, and the greater calculated value is used for pacing. Note that as many as four different calculation results may be taken into account for a concurrent R1 device with both mirrors operating in asynchronous mode (group-level pacing for each mirror, device-level pacing for each mirror), using the greatest calculated delay in the calculation.

Identifying devices that cannot be paced in a cascaded SRDF configuration

A source device might not be paced because it has been set exempt from group-level write pacing or because it is not currently pace-capable.

- Exempt source devices (R1 or R21) have been excluded from group-level write pacing using the `-rdfa_wpace_exempt` option of the `symrdf` command. Exempt devices can be paced by device-level write pacing.
- R21 devices (in an R21>R2 pair) are not pace-capable if the corresponding R1>R21 SRDF pair is read/write (RW) on the SRDF link and operating in an adaptive copy mode. A device that is not pace-capable cannot be paced by device-level write pacing or group-level write pacing. The `-force` option is required for actions that will cause a device to become not pace-capable.

To identify devices that cannot be paced:

1. Use the `symcfg list` command with the `-rdfa` option to determine if the SRDF/A session includes devices that cannot be paced. This command provides the following information related to write pacing:
   - The state of write pacing (group-level and device-level) for the SRDF group

- Whether write pacing is currently activated and supported
- Whether write pacing is configured for autostart
- Whether there are devices in the SRDF/A session that might not be paced either because they have been set exempt from group-level write pacing or because they are not pace-capable.

  To view write pacing information for SRDF group 153:

  symcfg list -sid 1134 -rdfg 153 -rdfa

```
Symmetrix ID : 000195701134

                 S Y M M E T R I X   R D F A   G R O U P S

-------- ---------- -------- ----- --- --- --------- ------------------------
                                                        Write Pacing
 RA-Grp  Group      Flags    Cycle Pri Thr Transmit   Delay  Thr GRP DEV FLGS
         Name       CSRM TDA time          Idle Time (usecs) (%) SAU SAU P
-------- ---------- -------- ----- --- --- --------- ------- --- --- --- ----
153 (98) lc153142   .IS- XI.    15  33  50 000:00:00   50000  60 I.- I.- X
.
.
    (FLGS) Flags for Group-Level and Device-Level Pacing:
      Devs (P)aceable : X = All devices, . = Not all devices, - = N/A
```

An X in the FLGS P column indicates that all of the devices in the SRDF group can be paced. A period in the FLGS P column indicates that some of the devices in the SRDF group cannot be paced either because they have been set exempt from group-level write pacing or because they are not pace-capable.

2. Use the `symrdf list` command to determine which devices cannot be paced.

   a. Use the `symrdf list` command with the `-rdfa_wpace_exempt` option to identify devices that are exempt from group-level write pacing.

   b. Use the `symrdf list` command with the `-rdfa_not_pace_capable` option to identify devices participating in the SRDF/A session that are not pace-capable.

3. Use the `symdev show` command to obtain additional information about the devices identified in the previous step. This command provides the following information related to write pacing:

   - Whether the device is exempt from group-level write pacing
   - Whether write pacing is currently activated and supported
   - Whether the device is pace-capable

     To view write pacing information for device 00d1:

     symdev show -sid 230 00d1

```
.
.
.
Write Pacing Information
            {
            Pacing Capable                     : Yes
            Configured Group-level Exempt State: Disabled
            Effective Group-level Exempt State : Enabled
            Group-level Pacing State           : Enabled
            Device-level Pacing State          : Disabled
```

```
.
.
.
```

Setting SRDF/A group-level write pacing attributes

Use the `symrdf set rdfa_pace` command to set the SRDF/A write pacing attributes for an SRDF group.

symrdf set rdfa_pace command syntax

```
symrdf -sid SymmID -rdfg GrpNum [-v] [-symforce]
     [-noprompt] [-i Interval] [-c Count]

.............

   set rdfa_pace
       [-dp_autostart {on | off}]
       [-wp_autostart {on | off}]
       [-delay 1 - 1000000]
       [-threshold 1 - 99]>
       [-both_sides]
```

To set these group attributes, the remote side must be reachable.

Where:

- `-dp_autostart` (`-dp_aut`) specifies whether SRDF/A device-level pacing is automatically enabled or disabled when an SRDF/A session is activated or deactivated for an SRDF group.

  Valid state values are on or off.

  Default is off.

- `-wp_autostart` (`-wp_aut`) specifies whether the SRDF/A group-level pacing feature is automatically enabled or disabled when an SRDF/A session is activated for an SRDF group.

  Valid state values are on or off.

  Default is off.

- `-delay` (`-del`) sets the maximum host I/O delay, in microseconds, that the SRDF/A write pacing can cause.

  Valid values are 1 through 1000000 microseconds.

  The default is 50000 microseconds.

- `-threshold` (`-thr`) sets the minimum percentage of the array write-pending cache at which the array begins pacing host write I/Os for an SRDF group.

  Valid values are between 1 and 99.

  The default is 60.

- `-both_sides` sets the SRDF/A write pacing attributes on both the source and target sides of an SRDF/A session. Otherwise, these attributes are only set on the source side.

Note: If you plan on swapping the personalities of the R1 and R2 devices, configure the same SRDF/A write pacing values on both sides.

Examples In the following example, SRDF/A group-level write pacing is enabled for SRDF group 12 with:

- A maximum of a 1000 microsecond delay
- A write pending cache threshold of 55 percent

  If the calculated delay is less than the specified delay (1000), then the calculated delay is used.

```
symrdf -sid 134 -rdfg 12 set rdfa_pace -delay 1000 -threshold 55 -wp_autostart on
```

Use the `-both_sides`, option to display two entries for each attribute being applied; one for the source side and one for the target side:

```
symrdf -sid 432 -rdfg 75 set rdfa_pace -delay 500 -threshold 10 -wp_autostart on
   -dp_autostart on -both_sides
```

Activating write pacing

symrdf activate command syntax

To activate and deactivate SRDF/A write pacing at the device-group level:

```
symrdf -g DgName [-v | -noecho] [-force] [-symforce]

   activate [-rdfa_dse | -rdfa_pace | -rdfa_wpace | -rdfa_devpace] |
   -rdfa_wpace_exempt [LdevName [LevdevName....]]

   deactivate [-rdfa_dse | -rdfa_pace | -rdfa_wpace | -rdfa_devpace]|
   -rdfa_wpace_exempt [LdevName [ LevdevName....]]
```

Examples To activate group-level write pacing for SRDF group 76:

```
symrdf -sid 123 -rdfg 76 activate -rdfa_wpace
```

To exempt `DEV001` in the `prod` group from SRDF/A write pacing:

```
symrdf -g prod -rdfg 76 -rdfa_wpace_exempt DEV001
```

To deactivate device-level write pacing for `DEV012` in the `prod` device group:

```
symrdf -g prod deactivate -rdfa_devpace DEV012
```

Controlling simultaneous group-level and device-level write pacing

When write pacing is activated at both group-level and device-level, Enginuity monitors both the SRDF link performance of the SRDF/A session and the performance of the devices on the R2 side.

Requirements for controlling group and device-level write pacing

- The `symrdf activate/deactivate -rdfa_pace` commands are targeted at all devices in the SRDF group.
- The R1 array is accessible.

- The SRDF/A session being controlled is active and contains at least one participating device.
- The `symrdf deactivate -rdfa_pace` command requires the following:
  - The R2 array is accessible to verify that there are no TimeFinder/Snap or TimeFinder/Clone sessions off the R2 devices before deactivating device-level pacing.
  - If the SRDF/A session is in the transmit idle state, issue `symrdf deactivate -rdfa_pace -symforce` from the R1 side.

Examples To activate group-level and device-level write pacing simultaneously for the `ConsisGrp` CG:

```
symrdf -cg ConsisGrp activate -rdfa_pace
```

To exempt `DEV001` in the `prod` group from both group-level and device-level write pacing:

```
symrdf -g prod -sid 55 -rdfg 76 -rdfa_pace_exempt DEV001
```

To deactivate both group-level and device-level write pacing on the devices in `DeviceFile2`:

```
symrdf -file DeviceFile2 -sid 55 -rdfg 2 deactivate -rdfa_pace
```

Displaying SRDF/A

Note: Output of srdf list and query commands varies depending on whether SRDF/A is in multi-cycle mode (HYPERMAX OS) or legacy mode (Enginuity 5876).

Display SRDF/A group information

Use the `symdg show DgName` command to display SRDF/A session status information.

Use the `symrdf -g DgName query -rdfa` command to display SRDF/A group information.

SRDF/A-capable devices in an SRDF group are considered part of the SRDF/A session. The session status is displayed as active or inactive, as follows:

- Active indicates the SRDF/A mode is activated and that SRDF/A session data is currently being transmitted in operational cycles to the R2.
- Inactive indicates the SRDF/A devices are either Ready or Not Ready on the link and working in their basic mode (synchronous, semi-synchronous, or adaptive copy).

Note: If the links are suspended or a split operation is in process, SRDF/A is disabled and will show a session status of Inactive.

List SRDF/A- capable devices

Use the `symrdf list -rdfa` command to list SRDF/A-capable devices (R1, R2 and R21 devices) that are currently configured in SRDF groups.

Note: SRDF/A-capable does not mean the device is actually operating in asynchronous mode, only that it is capable of doing so. There is no command that lists devices that are actually operating in asynchronous mode.

The device type is shown as *R1* for SRDF/A-capable devices on the R1 and type *R2* for SRDF/A-capable devices on the R2.

The *R21* device type represents a cascaded SRDF device configuration.

CHAPTER 5
Consistency Group Operations

This chapter describes the following topics:

- Overview 152
- Creating an SRDF consistency group 155
- Enabling and disabling SRDF consistency protection 158
- Deleting an SRDF consistency group 164
- Suspending SRDF consistency protection 165
- Using the msc_cleanup command 166
- Modifying consistency groups 168
- Consistency groups with a parallel database 180
- Consistency groups with BCV access at the target site 181

Overview

SRDF consistency preserves the dependent-write consistency of devices within a group by monitoring data propagation from source devices to their corresponding target devices. If a source R1 device in the consistency group cannot propagate data to its corresponding R2 device, SRDF consistency suspends data propagation from *all* the R1 devices in the group.

SRDF consistency allows you to quickly recover from certain types of failures or physical disasters by retaining a consistent, DBMS-restartable copy of your database.

SRDF consistency group protection is available for SRDF/S and SRDF/A.

An SRDF *consistency group* is a composite group comprised of SRDF devices with consistency enabled.

The devices in the consistency group are configured to act in unison to maintain the integrity of a database when distributed across multiple arrays or across multiple devices within an array.

Domino mode also ensures consistency of a remote database.

Consistency protection using the SRDF daemon

The SRDF daemon (storrdfd) provides consistency protection for:

- SRDF/A Multi-Session Consistency (MSC) consistency groups in multi-array environments
- SRDF/S RDF-Enginuity Consistency Assist (ECA) consistency groups in multi-array environments
- Multiple SRDF groups within the same array
- For MSC consistency groups, the SRDF daemon performs cycle switching and cache recovery for all SRDF/A sessions within a consistency group, and manages the R1 -› R2 commits for SRDF/A sessions in multi-cycle mode.

  If a data flow interruption (such as a trip event) occurs, storrdfd:

  - Halts R1-›R2 data propagation
  - Analyzes the status of all SRDF/A sessions.
  - Either commits the last cycle of data to the R2 targets or discards it.
- For RDF-ECA consistency groups, storrdfd continuously polls SRDF/S sessions for data flow interruptions.

  If any R1 device is unable to propagate data to its R2 target, storrdfd:

  - Halts all R1-›R2 data flow within an RDF-ECA consistency group.

storrdfd ensures that you always have a consistent R2 copy of a database at the point in time in which a data interruption occurs.

Before you begin

Before storrdfd can monitor and manage a consistency group, you must:

- Create a composite group with SRDF consistency enabled (`-rdf_consistency` option)
- Enable the composite group (`symcg enable` command).

"Creating an SRDF consistency group" on page 155 provides more information.

Enabling the SRDF daemon

The storrdfd daemon is required for SRDF consistency group operations.

By default, the storrdfd daemon is disabled and must be enabled for all applications using the SYMAPI configuration database file and SRDF consistency protection.

Enable storrdfd using the following SYMAPI options file setting:

```
SYMAPI_USE_RDFD=ENABLE
```

Each host running the SRDF daemon must also be running the base daemon (storapid).

EMC Solutions Enabler CLI Command Reference explains common daemon tasks, including how to start and stop daemons.

Enabling the Group Naming Services daemon

The storrdfd daemon runs on each host for which SRDF consistency is required.

If the Group Naming Services (GNS) daemon is enabled, storrdfd relies on GNS to propagate updated CG definitions to all hosts locally attached to the same set of arrays.

If GNS is not enabled, you must manually recreate the updated CG definition on each one of these hosts.

Enable GNS on each host using the following SYMAPI options file setting:

```
SYMAPI_USE_GNS=ENABLE
```

IMPORTANT

When using GNS, enabling the gns_remote_mirror option in the daemon_options file will not mirror the CG if it includes any devices listed in "Mirroring exceptions" in the *EMC Solutions Enabler Array Management CLI User Guide*.

Redundant consistency protection

You can run two instances of the SRDF daemon simultaneously on separate control hosts to create redundant consistency protection for composite groups.

Simultaneous SRDF daemons perform independent monitoring and switching operations. If one fails, the other SRDF daemon takes it place, and completes all pending tasks (commit the last cycle to the target site).

Redundant SRDF daemons allow you to avoid service interruptions caused by:

- Performance bottlenecks on one of the control hosts
- Link failures of the redundant SRDF daemons
- Failure of one control hosts

Each control host must have a common view of the composite group being monitored. To give each control host a common view, do one of the following:

- Run the GNS daemon on each control hosts, as shown in Figure 13, or
- Manually define the composite group on all control hosts.

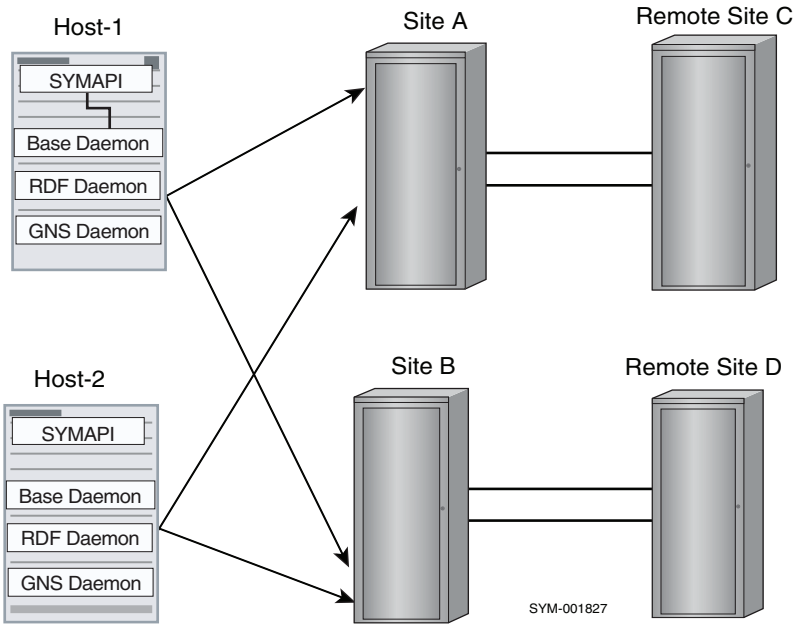


Figure 13 Running redundant hosts to ensure consistency protection

In Figure 13, Host-1 and Host-2 run all three daemons: base daemon, SRDF daemon, and GNS daemon to ensure data consistency protection

IMPORTANT

EMC strongly recommends running redundant SRDF daemons on at least two control hosts at each site. This ensures at least one SRDF daemon is available to perform time-critical, consistency monitoring operations.

EMC recommends that you do not run the SRDF daemon on the same control host running the database applications. Use this control host to issue other control commands (such as SRDF, TimeFinder, and Clone operations).

If the control host is powerful enough to efficiently handle all CPU operations, and is configured with sufficient gatekeeper devices for all your management applications, you can run ECC and Unisphere for VMAX with the Solutions Enabler daemons.

SRDF consistency group operations

SRDF composite groups are initially created using the `symcg create` command. Once they are created, they are populated with devices and device groups.

In order to be enabled as an SRDF consistency group, the composite group must be:

- Defined as a type RDF1, RDF2, or RDF21
- Have consistency enabled using the option`-rdf_consistency` option.

`symrdf` control operations can change a composite group. For example, a device personality swap operation can change an RDF1 CG to an RDF2 CG. SRDF control operations (`failover -establish` and `swap` operations) cannot change the type of an ANY composite group but can affect the devices in that CG.

"Consistency group control operations and applicable pair states" on page 396 provides a list of control actions and the required SRDF pair states for consistency group operations.

Note: Enginuity 5773.150 and higher is required for thin devices in a composite group in SRDF/S and SRDF/A configurations.

Creating an SRDF consistency group

The following steps illustrate how to build a consistency group when devices in the group are either all synchronous or all asynchronous.

Note: All devices containing application and array data must be included in the consistency group for each DBMS or across the DBMS controlling the multi-database transactions.

1. Use the `symcfg list` command to list all SRDF (RA) groups on the source arrays connected to the local hosts to determine which devices to include in the CG:

   ```
   symcfg list -rdfg all
   ```

2. Use the `symcg create` command to create a consistency group (`ConsisGrp)` on one of the local hosts.

   Specify the SRDF type of the group and the `-rdf_consistency` option:

   ```
   symcg create ConsisGrp -type rdf1 -rdf_consistency
   ```

3. Use the `symcg addall` command to add the devices from an SRDF (RA) group, such as `RDG 64`, into the consistency group (`ConsisGrp)`:

   ```
   symcg -cg ConsisGrp -sid 3264 addall dev -rdfg 64
   ```

4. In a database configuration with multiple local hosts, you must build the same consistency group on all local hosts in the configuration.

   You can use the `symcg export` command to manually transfer the consistency group definition, or if enabled, use GNS to automatically transfer it.

   The following commands create the `consisgrp.txt` text file containing the new `ConsisGrp` composite group definition and then transfer it to `Host-1`:

```
symcg export ConsisGrp -f consisgrp.txt
rcp consisgrp.txt Host-1:/.
```

In the following command, the `-rdf_consistency` option adds the imported `ConsisGrp` definition to the SRDF consistency database on `Host-1`:

```
symcg import ConsisGrp -f consisgrp.txt -rdf_consistency
```

5. Verify that all devices in the group are either all synchronous or all asynchronous.

```
symrdf -cg ConsisGrp verify -async
```

6. If the devices are currently operating with synchronous replication and you want them to be operating asynchronously, set the composite group for asynchronous replication:

```
symrdf -cg ConsisGrp set mode async
```

7. If the SRDF pairs are not in the Consistent or Synchronized state at this time (the Split or Suspended state), you can use the `symrdf establish` command to initiate SRDF copying of R1 data to the R2 side.

```
symrdf -cg ConsisGrp establish
```

The device state is SyncInProg until the Consistent or Synchronized state is reached.

With asynchronous replication, it may take two cycle switches for all devices to reach the Consistent state.

In multi-cycle mode, if either the link is or destaging the R2Apply cycle is slow, it may take more than 2 cycle switches for all devices to reach Consistent state.

8. From one of the local hosts, use the `symcg enable` command to enable the composite group for consistency protection:

```
symcg -cg ConsisGrp enable
```

The `ConsistGrp` CG becomes an SRDF consistency group managed by the SRDF daemon. The SRDF daemon watches for any problems with R1->R2 data within the `ConsisGrp` CG.

Creating composite groups from various sources

Topics in this section include:

- "Creating a composite group from an existing device group"
- "Creating a composite group from an RDBMS database"
- "Creating a composite group from a logical volume group"

Note: The E-Lab™ Interoperability Navigator at http://elabnavigator.EMC.com provides detailed interoperability information.

Creating a composite group from an existing device group

You can use the `symdg` command with the `-rdf_consistency` option to translate the devices of an existing device group to a new or existing composite group.

In the following example the `symdg` command:

- Translates devices to SRDF
- Adds all devices from a device group Symm64DevGrp to a composite group `ConsisGrp`.
- Adds the composite group to the SRDF consistency database on the host
- Enables the group for SRDF consistency protection:

```
symdg dg2cg Symm64DevGrp ConsisGrp -rdf_consistency
```

Creating a composite group from an RDBMS database

You can use the `export` command to translate the devices of an existing RDBMS database or tablespace to a new or existing composite group.

Note: For SYMCLI to access a specified database, you must set the `SYMCLI_RDB_CONNECT` environment variable to the username and password of the array administrator's account.

In the following example, a local connect is used, and the `export` command sets the variable to a username of "array" and a password of "manager".

```
export SYMCLI_RDB_CONNECT=array/manager
```

Note: The Bourne and Korn shells use the `export` command to set environment variables. The C shell uses the `setenv` command.

Connecting by network

When connecting by the network, you need to add a database-specific variable to the `RDB_CONNECT` definition.

When connecting through the network in an Oracle environment, Oracle has a network listener process running.

You must add an Oracle connection string such as the Transparent Network Substrate (TNS).

Examples

In the following example, the `export` command adds the TNS alias name "api217":

```
export SYMCLI_RDB_CONNECT=array/manager@api217
```

When connecting through the network in an SQL Server 2000 environment you must add a string to indicate the ODBC data source administrator.

In the following example, the `export` command adds string "HR":

```
set SYMCLI_RDB_CONNECT=array/manager@HR
```

Optionally, you can set the `SYMCLI_RDB_TYPE` environmental variable to a specific type of database (`oracle`, `informix`, `sqlserver`, or `ibmudb`) so that you do not have to include the `-type` option on the `symrdb rdb2cg` command line.

In the following example, the `export` command sets this variable to `oracle`:

```
export SYMCLI_RDB_TYPE=oracle
```

Translating devices

You can translate the devices in a database to a composite group.

Examples In the following example, the `symrdb rdb2cg` command:

- Translates the devices of an Oracle-type database named oradb to an RDF1 type composite group named `ConsisGrpDb`.
- The `-rdf_consistency` option adds the composite group to the SRDF consistency database on the host:

```
symrdb -type oracle -db oradb rdb2cg ConsisGrpDb -cgtype rdf1 -rdf_consistency
```

You can translate the devices in an Oracle type tablespace to a composite group.

In the following example, the `symrdb tbs2cg` command translates the devices of an oracle type tablespace orats to an RDF1 type composite group named `ConsisGrpTs`:

```
symrdb -type oracle -tbs orats tbs2cg ConsisGrpTs -cgtype rdf1 -rdf_consistency
```

With most RDBMS database arrays, you must set up environment variables specific to that array.

Oracle arrays use `ORACLE_HOME` and `ORACLE_SID`.

Sybase arrays use `SYBASE` and `DSQUERY`.

Creating a composite group from a logical volume group

You can use the `symvg` command to translate the devices of an existing logical volume group to a new or existing composite group. This command does not require environment variables.

Example In the following example, the `symvg` command:

- Translates the devices of a logical volume group named LVM4vg to an RDF1 type composite group named `ConsisGrp`.
- The `-rdf_consistency` option adds the composite group to the SRDF consistency database on the host:

```
symvg vg2cg LVM4vg ConsisGrp -cgtype rdf1 -rdf_consistency
```

Enabling and disabling SRDF consistency protection

You can enable or disable consistency protection for all the devices in a composite group. When you enable the composite group for consistency, the group is referred to as an SRDF consistency group.

Restrictions

- You can have either consistency protection or the domino effect mode enabled for a device, but not both.
- When a composite group is enabled for consistency protection:
  - Its name cannot be changed without first disabling the consistency protection. After the name change, re-enable the composite group using the new name.

- If the composite group is enabled for SRDF/A consistency protection, the SRDF daemon immediately begins cycle switches on the SRDF groups within the composite group (or named subset in MSC mode).

  The cycle switches for all SRDF groups will be performed at the same time. The interval between these cycle switches is determined by the smallest minimum cycle time defined on the R1 SRDF groups in the composite group (or named subset).

  The smallest minimum cycle time supported by the SRDF daemon is 3 seconds. This value is used if the smallest minimum cycle time across all component groups is less than 3 seconds.

◆ If you change the minimum cycle time for any of the R1 SRDF groups while the composite group (or named subset) is enabled for SRDF/A consistency protection, the new minimum cycle time will not take effect until you disable consistency protection and then re-enable it.

◆ If you change the minimum cycle time, the new minimum cycle time will not take effect until you disable consistency protection and then re-enable it.

◆ You can change contents of a composite group by doing one of the following:

- Disable consistency protection on a composite group while you add or remove devices, and then re-enable consistency protection after editing the composite group.

  Devices in the composite group are unprotected during the time required to edit and then re-enable the composite group.

- For RDF1 composite groups, you can dynamically modify the composite group while maintaining consistency protection during the editing process.

  "Modifying consistency groups" on page 168 provides more information.

Enabling consistency: composite group vs. SRDF group name

You can enable and disable consistency protection at the composite group level or at the SRDF group name level.

Composite group level

When consistency is enabled at the composite group level, all devices within the consistency group operate as a single unit.

If one R1 device in a CG is unable to propagate data to its R2 target, the SRDF links of all the devices within that CG are suspended.

To enable consistency protection at the composite group level, all device mirrors must be operating in the same SRDF mode: all device mirrors must be operating either synchronously or asynchronously.

Use the `symcg enable` and `symcg disable` commands to enable/disable consistency protection at the composite group level. All device pairs in the specified group are enabled/disabled.

Examples To enable consistency protection for all device pairs in composite group `prod` CG:

```
symcg -cg prod enable
```

To disable consistency protection for all device pairs in `prod` CG:

```
symcg -cg prod disable
```

SRDF group name level

When consistency protection is enabled at the SRDF group level, only the devices in the specified SRDF group operate as a unit.

If an R1 device in a CG cannot send data to its R2 target, the SRDF links for only those devices in the specified SRDF group of the CG are suspended.

SRDF group protection is useful for concurrent devices with one mirror operating in synchronous mode and the other mirror operating in asynchronous mode.

To enable consistency protection at the SRDF group name level, you must first define one or more named subsets of devices within the composite group.

A subset can consist of one or more of the SRDF groups within the composite group.

Restrictions

When a subset of a CG is enabled for consistency protection at the SRDF group name level:

- You must disable consistency protection on the subset before you can:
  - Change the name of the subset.
  - Add or remove SRDF groups to the subset.

    Note: For an RDF1 composite group, you can dynamically modify the contents of a subset while consistency protection is enabled. "Modifying consistency groups" on page 168 provides more information.

- You cannot enable a composite group at the CG level and a member SRDF group at the same time.
  - If a composite group is enabled at the CG level, no part of it can be simultaneously enabled at the SRDF group name level.
  - If a subset of the group is enabled at the SRDF group name level, the group cannot be enabled at the CG level.

Example

In the following example, composite group SALES consists of a set of concurrent SRDF devices distributed across two arrays, 076 and 077.

On array 076:

- SRDF group 100 operates in asynchronous mode, and
- SRDF group 120 operates in synchronous mode.

On array 077:

- SRDF group 101 operates in asynchronous mode, and
- SRDF group 121 operates in synchronous mode.

The following commands create two named subsets of the composite group:

- One containing the asynchronous SRDF groups:

```
symcg -cg SALES set -name sales1 -rdfg 76:100
symcg -cg SALES set -name sales1 -rdfg 77:101
```

- One containing the synchronous SRDF groups:

```
symcg -cg SALES set -name sales2 -rdfg 76:120
symcg -cg SALES set -name sales2 -rdfg 77:121
```

The following commands enable independent consistency protection for the two subsets:

```
symcg -cg SALES enable -rdfg name:sales1
symcg -cg SALES enable -rdfg name:sales2
```

SRDF/S devices

The `enable` action enables consistency protection either:

- Across all synchronous-mode devices in a consistency group, or
- Across all synchronous-mode devices in a named subset of a composite group.

If any R1 devices in an SRDF/S consistency group cannot propagate data to their corresponding R2 targets, the SRDF daemon suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets.

Examples

To enable consistency protection for SRDF/S pairs in the `prod` CG:

```
symcg -cg prod enable
```

To disable consistency protection for SRDF/S pairs in the `prod` CG:

```
symcg -cg prod disable
```

SRDF/A devices

The `enable` action enables consistency protection either:

- Across all asynchronous-mode devices in a consistency group, or
- Across all asynchronous-mode devices in a named subset of a composite group.

If an SRDF/A session that was enabled for consistency protection cannot propagate data from the R1 devices to their corresponding R2 target, Enginuity deactivates that session, suspending data propagation for all devices in the SRDF/A session and preserving R2 consistency.

If the consistency group or named subset of a composite group is comprised of multiple SRDF/A sessions, the SRDF daemon suspends data propagation for the other SRDF/A sessions, halting all data flow to the R2 targets in order to preserve R2 consistency.

Examples

To enable consistency protection for SRDF/A pairs in the `prod2` CG:

```
symcg -cg prod2 enable
```

To disable consistency protection for SRDF/A pairs in the `prod2` CG:

```
symcg -cg prod2 disable
```

Enabling SRDF consistency protection for concurrent SRDF devices

You can enable and disable consistency protection for concurrent devices at the composite group level or at the SRDF group name level.

Composite group level

To enable consistency protection for concurrent devices at the composite group level, all device mirrors must be operating in the same SRDF mode: all device mirrors must be operating either synchronously or asynchronously.

If the two groups are operating in asynchronous mode, they cycle-switch together.

In either asynchronous or synchronous mode, the SRDF daemon suspends the SRDF links for both groups if a concurrent R1 device is unable to propagate its data to either of its remote R2 partners. This preserves the consistency of R2 data.

Use the `symcg enable` and `symcg disable` commands to enable/disable consistency protection at the composite group level. All device pairs in the specified group are enabled/disabled.

If the concurrent mirrors are in asynchronous mode, the `enable` command enables consistency with MSC consistency protection.

If the concurrent mirrors are in synchronous mode, the `enable` command enables consistency with RDF-ECA consistency protection.

Examples

In the following example, composite group `prod` contains a concurrent R1 with two asynchronous target mirrors.

To enable consistency protection with MSC consistency protection for the two target mirrors:

```
symcg -cg prod enable
```

To disable consistency protection for all device pairs in `prod` CG:

```
symcg -cg prod disable
```

Group name level

When consistency is enabled at the SRDF group name level, the SRDF daemon monitors the SRDF groups separately.

If a concurrent R1 device is unable to propagate its data to one of its remote R2 partners, the daemon suspends the SRDF links for only the group representing that R2 mirror.

Restrictions

- If the two mirrors of the concurrent R1 devices in the composite group are operating in different modes (one mirror in synchronous mode and the other mirror in asynchronous mode), SRDF consistency protection cannot be enabled at the composite group level.

  You must individually enable each group representing the device mirrors by its group name.
- Enginuity 5874 or higher is required to enable RDF-ECA for both mirrors of a concurrent R1.
- Enginuity 5875 or higher is required to enable MSC for both mirrors of a concurrent R1.

- Table 14 lists the combinations of consistency protection modes allowed for the mirrors of a concurrent relationship.

Table 14 Consistency modes for concurrent mirrors

R1->R2 (first mirror)	R1->R2 (second mirror)
MSC	None
MSC	RDF-ECA
MSC	MSC
RDF-ECA	None
RDF-ECA	RDF-ECA
RDF-ECA	MSC
None	None
None	MSC
None	RDF-ECA

Example

1. Use the `symcg` command to define the group name to associate with the SRDF group number.

   In the following example, the name `cGrpA` is associated with SRDF group 55 on array 123:

   ```
   symcg -cg prod set -name cGrpA -rdfg 123:55
   ```

2. Use the `symcg` command to enable consistency protection for the SRDF group.

   In the following example, the name `cGrpA` is associated with SRDF group 55 on array 123:

   ```
   symcg -cg prod enable -rdfg name:cGrpA
   ```

   - If the mirrors in SRDF group 55 are operating in asynchronous mode, the SRDF group is enabled with MSC consistency protection.
   - If the mirrors in SRDF group 55 are operating in synchronous mode, the SRDF group is enabled with RDF-ECA protection.

3. Repeat the steps above to enable consistency protection for the second concurrent SRDF group

   Use a unique name for the second group.

Checking if device pairs are enabled for consistency protection

Use the `symrdf verify -enabled` command to validate whether device pairs are enabled for consistency protection.

"ping command" on page 39 provides more information.

Example

To verify whether the device pairs in the STAGING group are enabled for consistency protection:

```
symrdf -g STAGING verify -enabled
```

If none of the device pairs in the STAGING group are enabled for consistency protection, the following message displays:

```
None of the devices in the group 'STAGING' are 'Enabled'.
```

If all devices in the STAGING group were enabled for consistency protection, the following message displays:

```
All devices in the group 'STAGING' are 'Enabled'.
```

Use the `symrdf verify -enabled -synchronized -consistent` command to verify whether the device pairs are enabled for consistency protection and are in the synchronized OR consistent pair state.

Example To verify whether the device pairs in the STAGING group are enabled for consistency protection and are in the synchronized or consistent pair state:

```
symrdf -g STAGING verify -enabled -synchronized -consistent
```

If all devices are enabled and in the synchronized OR consistent pair state, the following message displays:

```
"All devices in the group 'STAGING' are 'Enabled' and in
'Synchronized, Consistent' states."
'Synchronized, Consistent' states."Blocking symcg enable on R2 side
```

Blocking symcg enable on R2 side

You can execute the `symcg enable` command from the R1 or R2 side of an SRDF relationship.

The `SYMAPI_ALLOW_CG_ENABLE_FROM_R2` in the options file allows you to prevent the `symcg enable` operation from being executed on the R2 side.

The default for `SYMAPI_ALLOW_CG_ENABLE_FROM_R2` is enabled. When enabled, this option allows the SDRF daemon running on the R2 side to close the RDF-ECA window due to a link failure, even though the failure prevents the R2 side from communicating with the R1 side.

This option can be set as:

- `ENABLE` – (Default) Allows the composite group to be enabled on the R2 side.
- `DISABLE` - Blocks the composite group from being enabled on the R2 side.

Deleting an SRDF consistency group

When you delete an SRDF consistency group from a CG, the SRDF daemon stops monitoring the CG.

⚠CAUTION

After deletion, SRDF consistency protection on the R2 data cannot be guaranteed even though the devices formerly in the CG may remain enabled.

Best practice is to disable consistency protection before deleting a group. "Enabling and disabling SRDF consistency protection" on page 158 provides more information.

The `-force` option is required if the group is disabled and there are members in the group.

The `-symforce` option is required if the group is enabled. The composite group remains enabled but is removed from the SYMAPI database.

Examples To delete a disabled SRDF consistency group `mycg1` (with members):

```
symcg delete mycg1 -force
```

Suspending SRDF consistency protection

You can use the `suspend`, `split` or `failover` commands to suspend consistency protection for all devices in an SRDF consistency group where all devices are either synchronous or asynchronous.

The state of the R2 devices at the end of the deactivation varies depending on whether the `suspend` or `split` command is used:

- `symrdf -cg suspend` - The R2 devices are in the write disabled state and cannot be accessed by the target-side hosts.

  R2 database copy is consistent with the production copy on the R1 side.

- `symrdf -cg split` - The R2 devices are enabled for both reads and writes by the target-side hosts.

The `-force` option is required:

When the same consistency group is defined on multiple hosts, you can initiate a suspend operation from any host provided the consistency group is enabled.

Note: If you execute the `failover` command on both mirrors of a concurrent R1 device, the concurrent R1 is converted into a concurrent R2 with a restore on both mirrors of the concurrent R2.

Examples To deactivate consistency in a consistency group named `ConsisGrp`:

```
symrdf -cg ConsisGrp suspend -force
```

To resume the SRDF links between the SRDF pairs in the SRDF consistency group and I/O traffic between the R1 devices and their paired R2 devices:

```
symrdf -cg ConsisGrp resume
```

Consistency protection is automatically restored upon resumption of the link.

Consistency protection is not disabled unless you specify `symcg -cg disable`.

For asynchronous replication, use the `symrdf -cg verify` command with the `-cg_consistent` option to ensure that the SRDF consistency group is SRDF-consistency enabled and in a consistent state.

A consistent state means that at least two cycle switches have occurred and all devices in each SRDF (RA) group have reached a consistent state.

Verify consistency

Example To verify that the SRDF consistency group ConsisGroup is SRDF-consistency enabled and in a consistent state.

```
symrdf -cg ConsisGrp verify -cg_consistent
```

(For synchronous operations) To verify if the device pairs in ConsisGroup are in Synchronized state:

```
symrdf -cg ConsisGrp verify -synchronized
```

Using the msc_cleanup command

When an SRDF/A *single* mode session is dropped, Enginuity automatically starts a cleanup process:

- The primary array marks new incoming writes as being owed to the secondary array.
- The capture and transmit delta sets are discarded, but the data is marked as being owed to the secondary array. All of these owed tracks are sent to the secondary array once SRDF is resumed, as long as the copy direction remains primary to secondary.
- The secondary array marks and discards the receive delta set only. Data is marked as tracks owed to the primary array.
- The secondary array makes sure the apply (N-2) delta set is safely applied to disk; this is the dependent-write consistent image.

When a SRDF/A *multiple* mode session with Multi-Session Consistency (MSC) is dropped, MSC cleanup operations either:

- Discards any incomplete SRDF/A data, or
- Commits completed data to the R2 to maintain dependent write consistency.

When a SRDF/A *multiple* mode session with MSC is dropped, additional cleanup is required in fault scenarios where all delta sets of a transition have not been fully applied or discarded.

Use the `msc_cleanup` command to cleanup after a session is dropped for devices operating in SRDF/A mode with consistency enabled MSC.

For example, if a link failure causes protection to be triggered, the daemon may not be able to process all cleanup operations for the R2 devices where the receive and apply delta sets reside. Run the `symrdf msc_cleanup` command manually from the R2 site. If no consistency group definition is available at the R2 site, direct the cleanup operation to an SRDF (RA) group that was included as part of the consistency group.

The command can be executed by composite group from the R1 or R2 site or by SRDF group from the R2 site.

Use the `symcfg list` command check whether a MSC cleanup operation is required.

Use the `symcfg list` command with the `-rdfg all` option to display whether a MSC cleanup operation is required for only SRDF (RA) groups on the specified array.

Output of the `symcfg list` command includes flag information for SRDF groups operating in SRDF/A mode. An X in the RDFA Flags “M" column denotes that an MSC cleanup operation is required.

Examples To cleanup a composite group (`mycg`):

```
symrdf -cg mycg msc_cleanup
```

To cleanup from the remote host at the R2 site for array 1123 and direct the command to SRDF group 4:

```
symrdf -sid 123 -rdfg 4 msc_cleanup
```

Modifying consistency groups

You can dynamically add or remove the following device types for an RDF1 consistency group without first disabling consistency protection:

- Simple R1
- Concurrent R11

Use the `symcg modify` command with the `add` and `remove` options to modify SRDF consistency groups.

Before you begin

Before you begin, you must understand how the SRDF daemon maintains consistency protection during dynamic modification:

- On the local host, the SRDF daemon continuously monitors the consistency group being changed.

  The SRDF daemon must be running locally on the host where the `symcg modify` command is issued.

- On other hosts, the SRDF daemons do the following:
  - On hosts running GNS - SRDF daemons monitor the consistency group as it is being modified as long as these hosts are locally attached to the same set of arrays as the control host.

    Depending on the timing of the GNS updates, there may be a brief period during which the SRDF daemon stops monitoring the consistency group while waiting for the updated consistency group definition to propagate to the local GNS daemon.

  - On hosts *not* running GNS - If the SRDF daemons are running Solutions Enabler versions lower than 7.3.1, the daemons stop monitoring the CG during dynamic modification. These older daemons see the old CG definition until the `symstar buildcg -update` command is issued.

    IMPORTANT

    EMC strongly recommends running GNS on your hosts to ensure consistency protection while dynamically modifying CGs.

    To restart monitoring the modified consistency group on these hosts, perform the following steps after you complete the modification operation:

    1. Use the `symcg export` operation on the control host to export the new definition to a file.
    2. Send the file to the other host.
    3. Use the `symcg import` operation on the control host to import the new definition file.

Restrictions

The following apply to dynamic `add` and `remove` options of the `symcg modify` command:

- Enginuity 5773 or higher is required on all arrays of the CG.

- All arrays are reachable.
- The SRDF daemon must be running locally on the host where the `symcg modify` command is issued.
- The `symcg modify` command only applies to RDF1 composite groups.

  It is not allowed for RDF2, RDF21, or type=ANY composite groups.
- The `symcg modify` command is not allowed for:
  - CGs consisting of device groups.
  - CGs containing concurrent SRDF devices.
  - Any devices in SRDF/Star mode.

    Use the `symstar modifycg` command to modify devices in the CG are in STAR mode. "SRDF/Star consistency group operations" on page 253 provides more information.
- The SRDF groups affected by the `symcg modify` command cannot contain any devices enabled for consistency protection by another CG.
- Devices within SRDF groups of the CG to be modified must be in one of the following SRDF pair states:
  - Synchronized
  - SyncInProg with invalid tracks owed to the R2
  - Consistent with no invalid tracks
  - Within an affected SRDF group, device pairs can be a mixture of Synchronized and SyncInProg or a mixture of Consistent and SyncInProg.

Note: If the `symcg modify` command fails, you can rerun the command or issue `symcg modify -recover`. No control operations are allowed on a CG until after a recover completes on that CG.

Preparing the staging area

Before you can dynamically modify SRDF consistency groups, you must create a staging area that mirrors the configuration of the CG. The staging area consists of:

- SRDF groups containing the device pairs to be added to a consistency group (`symcg modify -add` operations),
- SRDF groups for receiving the device pairs removed from a consistency group (`symcg modify -remove` operations).
- The SRDF groups in the staging area must be established between the same arrays as the SRDF groups in the consistency group.

  For concurrent CGs, the SRDF groups in the staging area must be established among three arrays.

Restrictions: SRDF groups and devices in the staging area

- SRDF groups cannot be part of an SRDF/Star configuration.

- Devices cannot be enabled for consistency protection.
- Devices cannot be defined with SRDF/Star SDDF (Symmetrix Differential Data Facility) sessions.
- BCVs are not allowed.
- All devices must be SRDF dynamic and of the same type:
  - Simple R1 devices
  - Concurrent R11 devices
- All device pairs must set in the same mode:
  - Adaptive copy disk
  - Adaptive copy write pending for diskless R21->R2 device pairs

  Note: Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS, and diskless R21 devices are not supported on arrays running HYPERMAX OS.

Restrictions: SRDF groups and devices for dynamic add operations

The dynamic modify add operation moves device pairs from the staging area into the SRDF groups of a consistency group.

All devices in the staging area must be in one of the following SRDF pair states for each SRDF group:

- Synchronized
- SyncInProg with invalid tracks owed to the R2
- Suspended
- Suspended with invalid tracks owed to the R2

If any device pair is Suspended (with or without invalid tracks on any of its SRDF groups), then the device pairs in the same SRDF group must all be Suspended.

Figure 14 shows a staging area for an R1->R2 configuration:

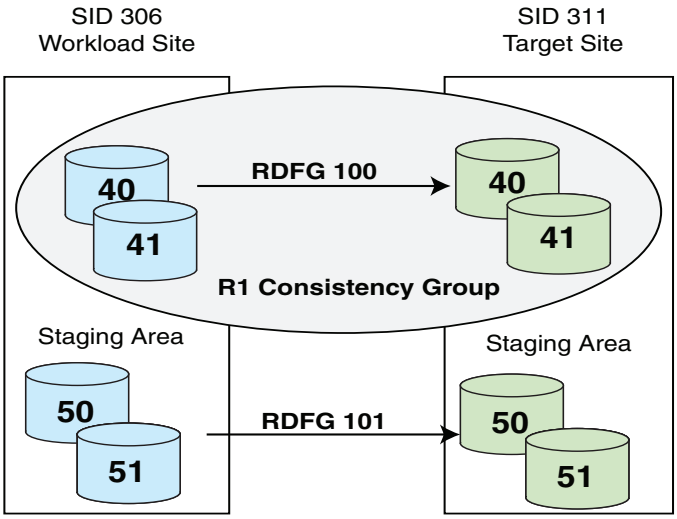


Figure 14 Staging area for adding devices to the R1CG consistency group

RDFG 101 is established between the same array as the RDFG 100 in the R1CG consistency group.

Figure 15 shows the R1CG consistency group after the dynamic add operation:.

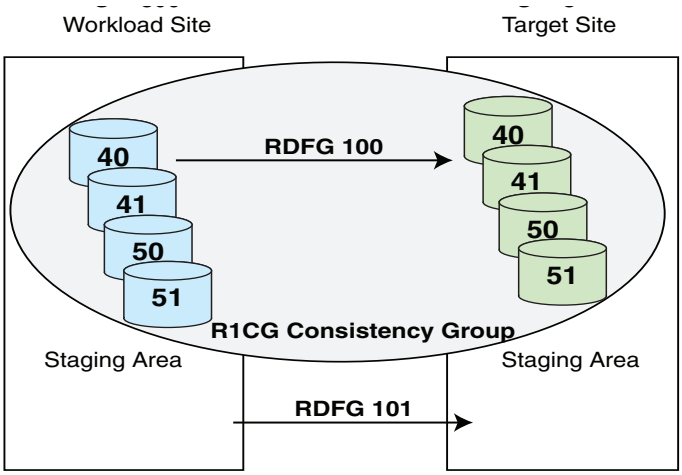


Figure 15 R1CG consistency group after a dynamic modify add operation

Devices 50 and 51 were moved to R1CG.

The staging area contains the empty RDFG 101.

Dynamic remove operation

The dynamic modify remove operation moves the device pairs from the consistency group into the SRDF groups in the staging areas.

To prepare the staging area for this operation, create the SRDF groups for receiving the device pairs removed from a consistency group.

Note: The dynamic modify remove operation must never leave an SRDF group empty.

Example Figure 16 shows empty group RDFG 34 configured to receive devices removed from RDFG 32:

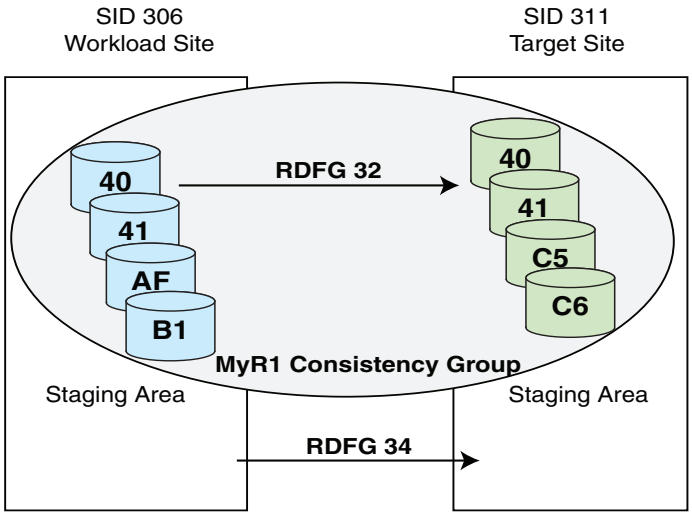


Figure 16 Preparing the staging area for removing devices from the MyR1 CG

The staging area consists of RDFG 34, an R1->R2 configuration established between the same array as RDFG 32 in the MyR1 consistency group.

Figure 17 shows the MyR1 consistency group and its staging area after the dynamic modify remove operation has completed.

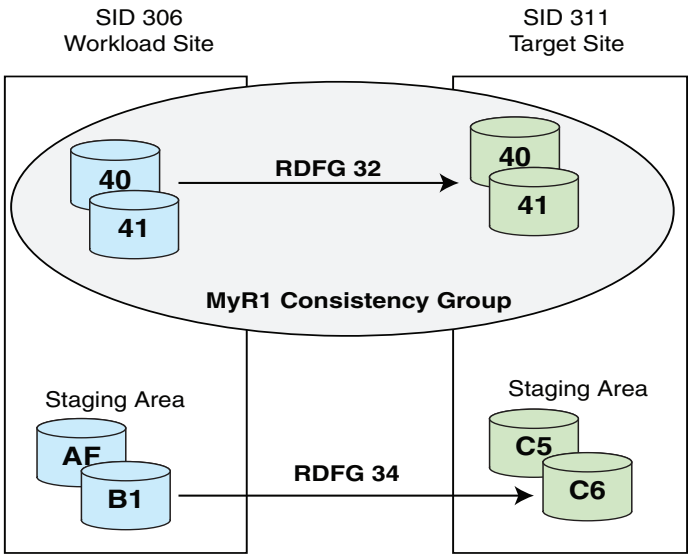


Figure 17 MyR1 CG after a dynamic modify remove operation

Restrictions - add operations

The following are restrictions for dynamically adding devices to an SRDF consistency group using the `symcg modify -add` command:

- The `symcg modify -add` command:
  - Cannot add new SRDF groups to the CG.
  - Cannot add a concurrent R11 device to a CG enabled at the composite group level.
  - Prohibits adding both mirrors of a concurrent R11 device to the same SRDF group name.
  - Cannot add a triangle of devices to a CG. In other words, a concurrent R11 device cannot have one R1 mirror paired with an R21 device, which is then paired with an R22 device that is paired with the other R1 mirror of the concurrent R1 device.
  - Prohibits adding a cascaded R1 device to a concurrent CG.
  - Prohibits adding a concurrent R1 device to a cascaded CG.
- If the target is a cascaded CG, the operation must be enabled by CG hop 1 or by the SRDF group name hop 1.
- If the target is a cascaded CG and the devices to be added are simple R1 devices, the CG cannot be enabled by CG hop 2 or by SRDF group name hop 2.
- If the target is a cascaded CG and the devices to be added are cascaded R1 devices paired with diskless R21 devices, then all R21 devices in the affected SRDF group must also be diskless.
- If the target is a cascaded CG and the devices to be added are cascaded R1 devices paired with non-diskless R21 devices, then all R21 devices in the affected SRDF group must be non-diskless.

Restrictions - remove operations

The following are restrictions for dynamically removing devices from an SRDF consistency group using the `symcg modify -remove` command:

- The dynamic modify remove operation must never leave an SRDF group empty.
- The `symcg modify -remove` command cannot remove SRDF groups from a consistency group.
- The `symcg modify -remove` command prohibits a cascaded R1 device from being removed from a consistency group enabled at the composite group level.
- The `symcg modify -remove` command cannot remove both legs of a concurrent R11 device if they are enabled for consistency protection by the same SRDF group name.

Adding devices to an RDF1 consistency group

Table 15 lists the allowable device types for a dynamic modify add operation on a composite group enabled for consistency protection at the composite group level and the SRDF group name level. This RDF1 CG is not concurrent or cascaded.

Table 15 Allowable device types for adding devices to an RDF1 CG

Device type in staging area	Enabled at CG level	Enabled at SRDF group name level
Simple R1 (R1->R2)	Allowed	Allowed
Concurrent R11	Not allowed	Only allowed if both affected SRDF groups in the CG already exist and are assigned to different SRDF group names.
Cascaded R1	Not allowed	Not allowed

Examples To move devices 50 and 51 from SRDF group 101 in the staging area to SRDF group 100 in `R1CG` on array 306:

```
symcg -cg R1CG modify -add -sid 306 -stg_rdfg 101 -devs 50:51 -cg_rdfg 100
```

To check if the devices were added to R1CG:

```
symrdf -cg R1CG query -detail
```

Adding devices to a concurrent RDF1 consistency group

Before you perform this procedure, review "Enabling SRDF consistency protection for concurrent SRDF devices" on page 161.

Table 16 shows the allowable device types for a dynamic modify add operation on a concurrent RDF1 composite group enabled for consistency protection at the composite group level and the SRDF group name level.

Table 16 Allowable device types for adding devices to a concurrent RDF1 CG

Device type in staging area	Enabled at CG level	Enabled at SRDF group name level
Simple R1 (R1->R2)	Allowed	Allowed
Concurrent R11	Not allowed	Only allowed if each mirror is assigned to a different SRDF group
Cascaded R1	Not allowed	Not allowed

Table 17 lists the allowable consistency modes for the SRDF groups of a concurrent CG.

Table 17 Supported consistency modes for concurrent SRDF groups

SRDF group 1 (first mirror)	SRDF group 2 (second mirror)
RDF-ECA	RDF-ECA
RDF-ECA	MSC
RDF-ECA	Not enabled

Table 17 Supported consistency modes for concurrent SRDF groups

SRDF group 1 (first mirror)	SRDF group 2 (second mirror)
Not enabled	RDF-ECA
MSC	RDF-ECA
MSC	MSC
MSC	Not enabled
Not enabled	MSC

Example In this example, device 20 is added to two independently-enabled SRDF groups of a CG.

Figure 18 on page 175 shows the staging area shared by array 306, 311, and 402 in a concurrent SRDF configuration:

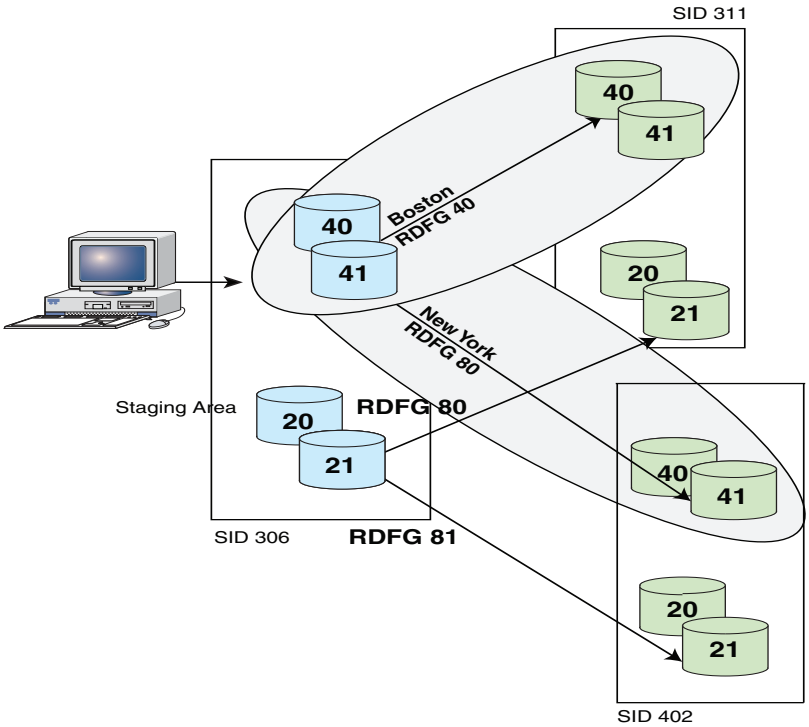


Figure 18 Adding a device to independently-enabled SRDF groups of a concurrent CG

The staging area contains devices 20 and 21.

Since the SRDF groups 70 and 71 of `ConCG` operate in different SRDF modes, they were enabled independently for consistency protection using the following SRDF group names:

- *Boston*: device pairs operate in SRDF/S mode and are set for RDF-ECA consistency protection.
- *New York*: device pairs operate in SRDF/A mode and are enabled for MSC consistency protection

To add only device 20 from the staging area into SRDF groups 70 and 71 of `ConCG`:

```
symcg -cg ConCG modify -add -sid 306 -stg_rdfg 80,81 -devs 20 -cg_rdfg 70,71
```

To check if the devices were added to `ConCG`:

```
symrdf -cg ConCG query -detail
```

Adding devices to a cascaded RDF1 consistency group

Before you perform this procedure, review "Checking if device pairs are enabled for consistency protection" on page 163.

Table 18 shows the allowable device types for a dynamic modify add operation on a cascaded R1 composite group enabled for consistency protection at the composite group level and the SRDF group name level.

Table 18 Allowable device types for adding devices to a cascaded RDF1 CG

Device type in staging area	Enabled at CG level			Enabled at SRDF group name level		
	Hop 1 enabled Hop 2 not enabled	Hop 1 enabled Hop 2 enabled	Hop 1 not enabled Hop 2 enabled	Hop 1 enabled Hop 2 not enabled	Hop 1 enabled Hop 2 enabled	Hop 1 not enabled Hop 2 enabled
Simple R1 (R1->R2)	Allowed	Not allowed	Not allowed	Allowed	Not allowed	Not allowed
Concurrent R11	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed
Cascaded R1	Allowed	Allowed	Not allowed	Allowed	Allowed	Not allowed

Table 19 lists the allowable consistency modes for the hops of a cascaded CG.

Table 19 Supported consistency modes for cascaded hops

R1->R21 (hop 1)	R21->R2 (hop 2)
RDF-ECA	MSC
RDF-ECA	Not enabled
MSC	Not enabled

Example Figure 19 shows a cascaded SRDF configuration sharing the staging area among array 306, 311, and 402:

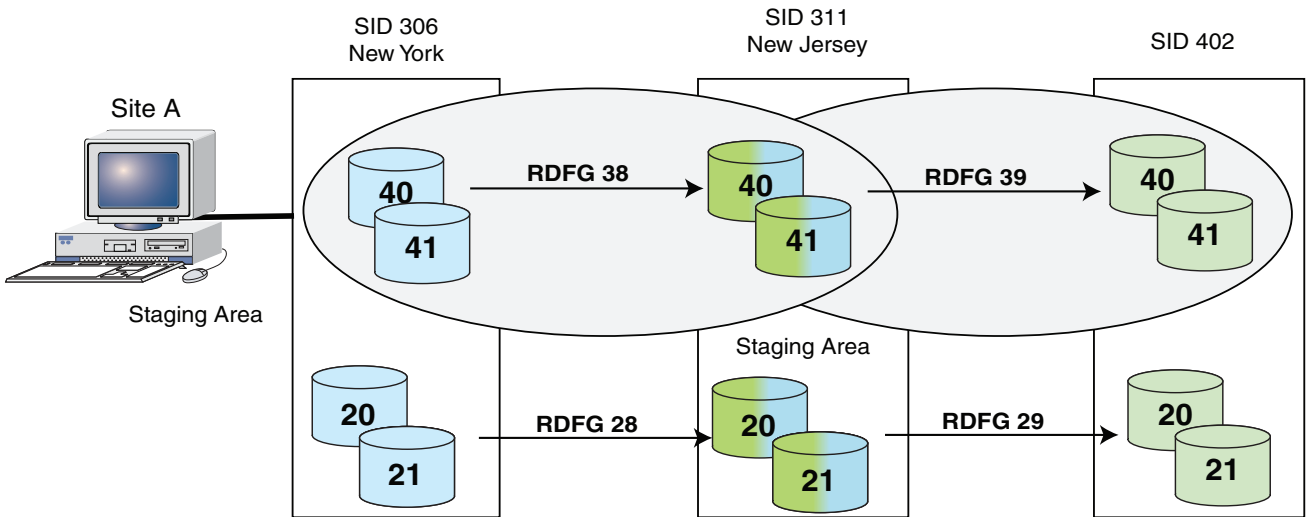


Figure 19 Adding devices to independently-enabled SRDF groups of a cascaded CG

.The staging area contains devices 20 and 21 to be added to `CasCG`.

The hops were independently enabled for consistency protection using the following SRDF group names:

- *New York*: device pairs operate in SRDF/S mode and are set for RDF-ECA consistency protection.
- *New Jersey*: device pairs operate in SRDF/A mode and are enabled for MSC consistency protection.

To add devices 20 and 21 from the staging area into SRDF groups 38 and 39 of `CasCG`:

```
symcg -cg CasCG modify -add -sid 306 -stg_rdfg 28 -devs 20:21 -stg_r21_rdfg 29
   -cg_rdfg 38 -cg_r21_rdfg 39
```

To check if the devices were added to `CasCG`:

```
symrdf -cg CasCG query -detail -hop2
```

Removing devices from an RDF1 consistency group

Table 20 shows the allowable device types for a dynamic modify remove operation on a composite group enabled for consistency protection at the composite group level and the SRDF group name level. This RDF1 CG is not concurrent or cascaded.

Table 20 **Allowable device types for removing devices from an RDF1 CG**

Device type in CG	Enabled at CG level	Enabled at SRDF group name level
Simple R1 (R1->R2)	Allowed	Allowed
Concurrent R11	Not applicable	Not applicable
Cascaded R1	Not applicable	Not applicable

Example To remove devices `50` and `51` from `RDFG 100` of `R1CG` on array 306 to `RDFG 101` in the staging area:

```
symcg -cg R1CG modify -remove -sid 306 -stg_rdfg 101 -devs 50:51 -cg_rdfg 100
```

Removing devices from a concurrent RDF1 consistency group

Table 21 shows the allowable device types for a dynamic modify remove operation on a concurrent R1 composite group enabled for consistency protection at the composite group level and the SRDF group name level.

Table 21 Allowable device types for removing devices from a concurrent RDF1 CG

Device type in CG	Enabled at CG level	Enabled at SRDF group name level
Simple R1 (R1->R2)	Allowed	Allowed
Concurrent R11	Not allowed	Only allowed if both mirrors are not enabled by the same SRDF group name.
Cascaded R1	Now allowed	Not allowed

Example To remove devices `20` through `30` from SRDF groups `70` and `80` of `ConCG` on array 306 into SRDF groups `71` and `81` in the staging area:

```
symcg -cg ConCG modify -remove -sid 306 -stg_rdfg 71,81 -devs 20:30 -cg_rdfg 70,80
```

Removing devices from a cascaded RDF1 consistency group

Table 22 shows the allowable device types for performing a dynamic modify remove operation on a cascaded R1 composite group enabled for consistency protection at the CG level and the SRDF group name level.

Table 22 Allowable device types for removing devices from a cascaded RDF1 CG

Device type in CG	Enabled at CG level			Enabled at SRDF group name level		
	Hop1 enabled Hop2 not enabled	Hop1 enabled Hop2 enabled	Hop1 not enabled Hop2 enabled	Hop1 enabled Hop2 not enabled	Hop1 enabled Hop2 enabled	Hop1 not enabled Hop2 enabled
Simple R1 (R1->R2)	Allowed	Not applicable	Not applicable	Allowed	Not applicable	Not applicable
Concurrent R11	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed
Cascaded R1	Allowed	Allowed	Not allowed	Allowed	Allowed	Not allowed

Example To remove device 20 of SRDF groups 38 (R1->R21) and 39 (R21->R2) of `CasCG` on array 306 into SRDF groups 28 and 29 in the staging area:

```
symcg -cg CasCG modify -remove -sid 306 -cg_rdfg 38 -devs 20 -cg_r21_rdfg 39 -stg_rdfg 28
-stg_r21_rdfg 29
```

Recovering from a failed dynamic modify operation

Details about dynamic modify operations (target CG, SRDF groups, staging area, and operation type) are stored in the Symmetrix File System (SFS).

If a dynamic modify operation fails and all sites are reachable:

1. Re-run the command with the *exact* parameters.

2. If the command fails again, execute the `symcg modify -recover` command:

   symcg modify -cg *CasCG* -recover

This command uses the dynamic modify command information in SFS.

The recover operation either:

- Completes the unfinished steps of the dynamic modify operation, or
- Rolls back any tasks performed on the CG before failure, placing the CG into its original state.

For example, if a concurrent R11 loses a link to one of its mirrors during a dynamic modify add operation, the recover operation may remove all devices added to the CG by this operation. This ensures that the CG device pairs are consistent at all three sites.

Consistency groups with a parallel database

Figure 20 illustrates an SRDF consistency group with a parallel database such as Oracle Parallel Server (OPS).

- The production database array spans two hosts and two arrays, A and C.
- A SRDF consistency group includes R1 devices from arrays A and C.

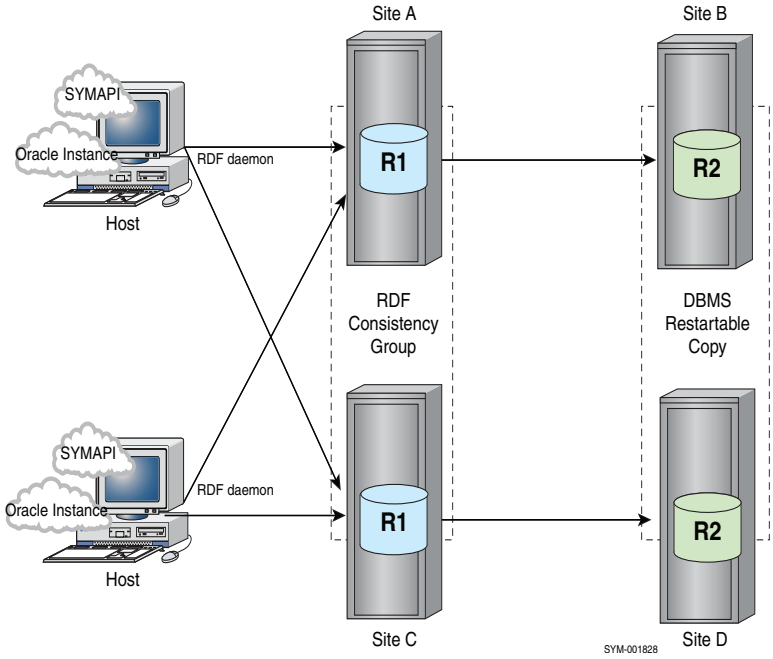


Figure 20 Using an SRDF consistency group with a parallel database configuration

The same consistency group definition must exist on both hosts. If enabled, Group Name Services (GNS) automatically propagates a composite group definition to the arrays and to all locally-attached hosts running the GNS daemon.

Although each production host can provide I/O to both R1 devices in the configuration, the DBMS has a distributed lock manager that ensures two hosts cannot write data to the same R1 device at the same time.

The SRDF links to two remote arrays (B and D) enable the R2 devices on those arrays to mirror the database activity on their respective R1 devices.

A typical remote configuration includes a target-side host or hosts (not shown in the illustration) to restart and access the database copy at the target site.

Figure 20 shows the SRDF daemons located on the production hosts.EMC recommends that you do not run the SRDF daemon on the same control host running database applications. "Redundant consistency protection" on page 153 provides more information.

Consistency groups with BCV access at the target site

When an SRDF consistency group includes devices on one or more source arrays propagating production data to one or more target arrays, TF BCVs at the target site can be indirectly involved in the consistency process.

Figure 21 illustrates a configuration with target-side BCVs that mirror the R2 devices:

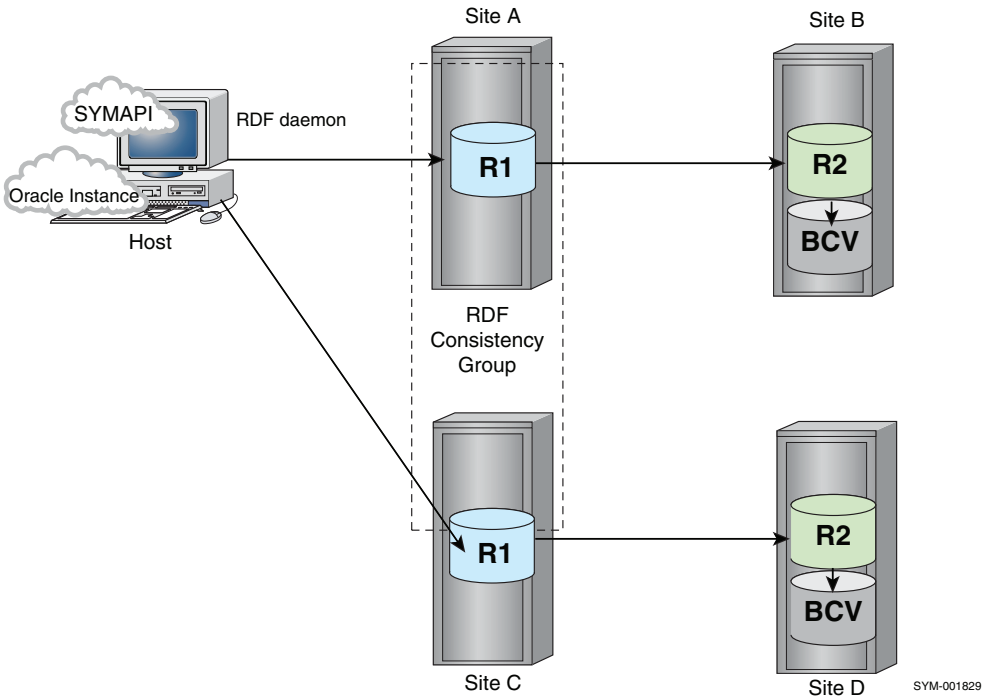


Figure 21 Using an SRDF consistency group with BCVs at the target site

You must split the BCV pairs at the target sites to access data on the BCVs from the target-side hosts.

The recovery sequence in a configuration that includes BCVs at the target site is the same as described in "Recovering from a failed dynamic modify operation" on page 178 with the following exception:

At the end of the sequence, the DBMS-restartable copy of the database exists on the target R2 devices and on the BCVs if the BCVs were synchronized with the target site's R2 devices at the time the interruption occurred.

When data propagation is interrupted, the R2 devices of the suspended SRDF pairs are in a Write Disabled state. The target-side hosts cannot write to the R2 devices, thus protecting the consistent DBMS-restartable copy on the R2 devices.

You can perform disaster testing and business continuance tasks by splitting off the BCV version of the restartable copy, while maintaining an unchanged R2 copy of the database. The R2 copy can remain consistent with the R1 production database until normal SRDF mirroring between the R1 and R2 sides resumes.

This configuration allows you to split off and access the DBMS-restartable database copy on the BCVs without risking the data protection that exists on the R2 devices when propagation of data is interrupted.

To manage the BCVs from the R2 side, associate the BCVs with a single SRDF consistency group defined on the target-site host that is connected to arrays B and D.

Figure 21 shows the SRDF daemons located on the production hosts.EMC recommends that you do not run the SRDF daemon on the same control host running database applications. "Redundant consistency protection" on page 153 provides more information.

CHAPTER 6
Concurrent Operations

This chapter describes the following topics:

- Overview 184
- Other documentation 186
- Creating/establishing concurrent SRDF devices 186
- Splitting concurrent SRDF devices 187
- Restoring concurrent devices 187
- Viewing concurrent SRDF devices 189

Overview

In a concurrent SRDF configuration, the source R1 device is mirrored to two R2 devices on two different remote arrays.

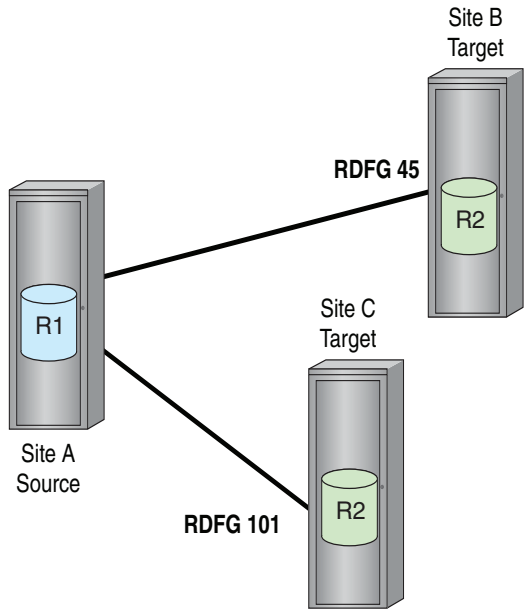


Figure 22 Concurrent SRDF

The two R2 devices operate independently but concurrently using any combination of SRDF modes.

Note: Starting with Enginuity 5875, both legs of the concurrent SRDF configuration can be in asynchronous mode

If both R2 mirrors are synchronous:

- A write I/O from the host at the R1 device side is returned as completed when both remote array' signal that the I/O is in cache at the remote side.

If one R2 is synchronous and the other R2 is adaptive copy:

- I/O from the R2 operating in synchronous mode must present ending status to the sending array before a second host I/O can be accepted.

  The host does not wait for the R2 operating in adaptive copy mode.

Restrictions

- The R2 devices at each remote array must belong to a different SRDF group.
- Asynchronous SRDF to both R2 devices requires Enginuity 5875 or higher on the R1 side.
- Simultaneous restore from both R2 devices to the R1 device cannot be performed.
- SRDF swap cannot be performed.

Both R2 devices in synchronous mode

If both R2 devices are in synchronous mode, both target sites have exact replicas of the source data. For this configuration, all three sites must be within synchronous distances.

Figure 23 shows three sites that are within synchronous distance:

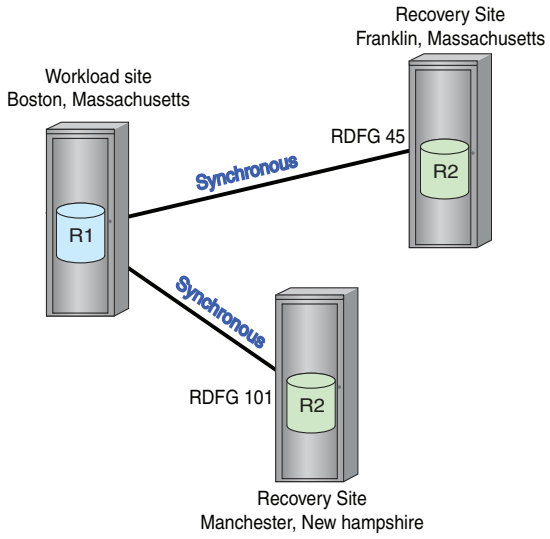


Figure 23 Concurrent SRDF/S to both R2 devices

Both R2 devices in asynchronous mode

You can configure concurrent SRDF/A to asynchronously mirror to recovery sites located at extended distances from the workload site.

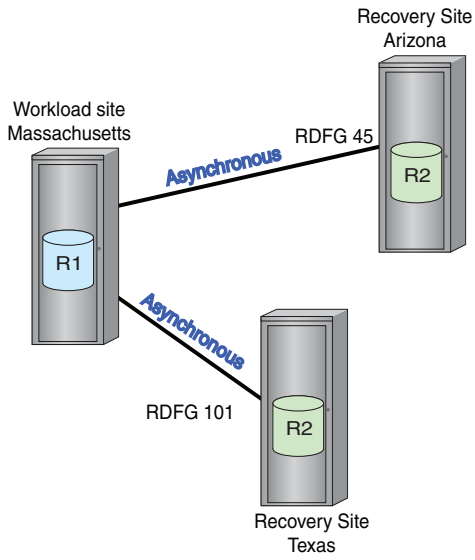


Figure 24 Concurrent SRDF/A to both R2 devices

With concurrent SRDF, you can build a device group or a composite group containing devices that only belong to the two SRDF groups representing the concurrent remote mirrors.

Your device group can also include BCV devices and SRDF devices that are not concurrent SRDF devices but that belong to either one of the concurrent SRDF groups.

Each mirror in a concurrent relationship must belong to a different SRDF group. When controlling or setting concurrent SRDF devices:

- `-rdfg n` performs the operation on the specified SRDF group number (remote mirror)
- `-rdfg ALL` performs the operation on the both SRDF groups.

Other documentation

Applicable pair states for concurrent SRDF operations

You can perform a control operation on one of these legs only if the other leg is in an acceptable pair state.

"Concurrent SRDF control operations and applicable pair states" on page 390 provides more information.

Consistency protection

You can enable consistency protection for devices in a concurrent configuration.

"Enabling SRDF consistency protection for concurrent SRDF devices" on page 161 provides more information.

Note: Consistency protection for two mirrors participating in different consistency groups with SRDF/S requires Enginuity 5874 and higher.

Concurrent SRDF operations

To configure a concurrent SRDF relationship:

1. Create the initial R1 -> R2 pair between the first array and second array.
2. Create the R11 -> R2 pair between first array and the third array.

Creating/establishing concurrent SRDF devices

Procedure

To create a device group for the concurrent SRDF devices and initially synchronize (establish) the devices across the concurrent SRDF links:

1. Use the `symdg` command to create an R1 device group.

```
symdg [-i Interval] [-c Count] [-v]
.....
create DgName -type RDF1
```

```
symdg create ConcGrp -type RDF1
```

2. Use the `symdg add` command to add all concurrent SRDF devices to the device group:

```
symdg -g DgName [-i Interval] [-c Count] [-v]
.....
add dev SymDevName
```

```
symdg add dev 0001 -g ConcGrp -sid 0001
```

```
symdg add dev 0021 -g ConcGrp
symdg add dev 002A -g ConcGrp
```

3. Use the `symrdf establish` command to establish concurrent SRDF pairs that belong to the device group for the first R2 devices:

   ```
   symrdf -g DgName [-v | -noecho]
   ......
   -rdfg GrpNum establish
   ```

   ```
   symrdf -g ConcGrp establish -rdfg 1
   ```

4. Repeat Step 3 to establish concurrent SRDF pairs that belong to the device group for the second R2 devices:

   ```
   symrdf -g ConcGrp establish -rdfg 2
   ```

   You can also use the `-rdfg ALL` option to simultaneously establish both mirrors of each SRDF pair in one command:

   ```
   symrdf -g concGrp -full establish -rdfg ALL
   ```

   Note: Business Continuance Volume (BCV) devices cannot contain concurrent SRDF mirrors.

Splitting concurrent SRDF devices

Use the `symrdf split` command to split concurrent SRDF pairs, either one at a time or at the same time.

Examples To split the concurrent pairs one at a time:

```
symrdf -g DgName split -rdfg GroupNum of first mirror
symrdf -g DgName split -rdfg GroupNum of second mirror
```

For example, to split the concurrent pairs for device group `concGrp` one at a time:

```
symrdf -g concGrp split -rdfg 1
symrdf -g concGrp split -rdfg 2
```

To split the concurrent pairs simultaneously:

```
symrdf -g DgName split -rdfg All
```

For example, to split the concurrent pairs for device group `concGrp` at the same time:

```
symrdf -g concGrp split -rdfg ALL
```

Note: Concurrent R1 devices can have two mirrors participating in different consistency groups with MSC consistency protection enabled. This applies only to Enginuity 5875 and higher.

Restoring concurrent devices

In concurrent configuration, there are two RDFG groups of R2 devices.

- You can restore the R1 device from either of the R2 devices.

  To restore the R1 device from either of the R2 devices, you must specify which R2 device to use.

- You can restore both the R1 and one R2 device from the second R2 device.

Restoring R1 from a concurrent R2

Use the `restore` command to restore only the R1 device from the specified R2:

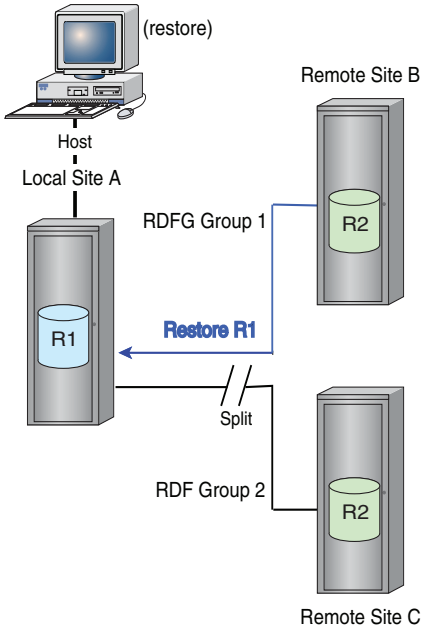


Figure 25 Restoring the R1 a concurrent configuration

When the `restore` command is executed:

- Both remote mirrors are split.
- The R1 device is restored from and synchronized with the R2 device in the specified RDFG group specified in the command.
- The R2 device belonging to SRDF group *not* used in the restore operation remains in the split state.

Use the `symrdf restore` command to restore from the specified RDFG group:

```
symrdf -g DgName restore -rdfg GroupNum of selected R2 mirror
```

Examples

To restore devices in group `concGrp` from RDFG group 1:

```
symrdf -g concGrp restore -rdfg 1
```

Use the `establish` command to re-establish the R2 devices not used in the restore operation:

```
symrdf -g DgName restore -rdfg GroupNum of group not used to restore
```

To re-establish second mirror (RDFG 2) for group `concGrp`:

```
symrdf -g concGrp establish -rdfg 2
```

Restoring both R1 and R2 from the second concurrent R2

Use the `restore` command with the `remote` option to restore both the R1 devices and the R2 devices on one leg from the R2 devices on the second leg:

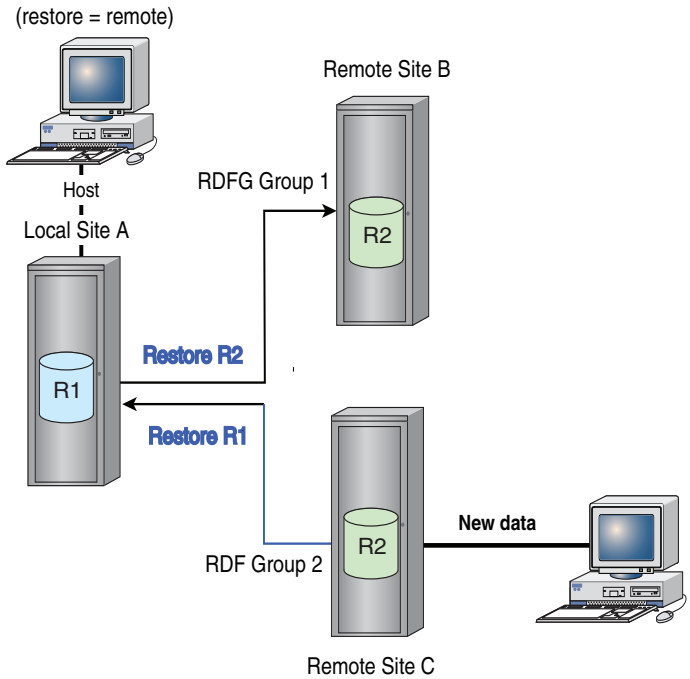


Figure 26 Restoring the source device and mirror in a concurrent SRDF configuration

When the `restore` command with the `remote` option is executed:

- Data from the specified R2 SRDF group 2 propagates data to the R1.
- The R1 SRDF group uses this data to restore the other R2 mirror, synchronizing all concurrent SRDF mirrors.

Note: You cannot simultaneously restore from both remote mirrors to the R1 device.

Use the `symrdf restore` command with the `remote` option to restore both the R1 devices and R2 devices on the second leg from the specified RDFG group:

```
symrdf -g DgName restore -rdfg GroupNum -remote
```

Example To restore the both the R1 and the R2 devices in RDF group 1 using the data in RDF group 2:

```
symrdf -g ConcGrp restore -rdfg 2 -remote
```

Viewing concurrent SRDF devices

Use the `symrdf list` command with the `-concurrent` option to display concurrent SRDF devices on the local array.

Each device of a concurrent pair belongs to a different RDF group, as shown in the RDF Typ:G column.

```
symrdf list -concurrent -sid 321

Symmetrix ID: 000192600321

                              Local Device View
-------------------------------------------------------------------------------
```

```
                        STATUS     MODES                      RDF  S T A T E S
Sym          RDF        ---------  -----  R1 Inv   R2 Inv ----------------------
Dev   RDev   Typ:G      SA RA LNK  MDATE  Tracks   Tracks Dev RDev Pair
---- ---- --------      ---------  ----- -------  ------- --- ---- -------------

00060 00060   R1:128    RW RW RW   S..1.       0        0 RW  WD   Synchronized
      00060   R1:228    RW RW RW   S..1.       0        0 RW  WD   Synchronized
00061 00061   R1:128    RW RW RW   S..1.       0        0 RW  WD   Synchronized
               .
               .
               .
```

Use the `query -rdfg all` command to display the state of concurrent SRDF pairs.

In the following example, concurrent SRDF pairs are in the process of synchronizing (SyncInProg):

```
symrdf -g conrdf query -rdfg all

Device Group (DG) Name             : conrdf
DG's Type                          : RDF1
.
.
.
Source (R1) View                   Target (R2) View         MODES
--------------------------------   ------------------------ ----- ------------
                ST                 LI      ST
Standard        A                  N       A
Logical         T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev     E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDAE  STATE
-------------------------------- -- ------------------------ ----- ------------

DEV001  00060 RW       0    69030 RW 0060 WD       0        0 S...  SyncInProg
              RW       0    69030 RW 0060 WD       0        0 S...  SyncInProg
DEV002  00061 RW       0    69030 RW 0061 WD       0        0 S...  SyncInProg
              RW       0    69030 RW 0061 WD       0        0 S...  SyncInProg
DEV003  00062 RW       0    69030 RW 0062 WD       0        0 S...  SyncInProg
```

During synchronization, use the `symrdf verify -summary` command to displays a summary message every 30 seconds until both concurrent mirrors of each SRDF pair are synchronized:

```
symrdf -g conrdf verify -summary -rdfg all -i 30 -synchronized
.
.
None of the devices in the group 'conrdf' are in 'Synchronized'
state.
.
.
Not All devices in the group 'conrdf' are in 'Synchronized' state.
.
.
All devices in the group 'conrdf' are in 'Synchronized' state.
```

CHAPTER 7
Cascaded Operations

This chapter describes the following topics:

- Overview 192
- Setting up cascaded SRDF 194
- Managing R21 devices 197
- Setting up cascaded SRDF with EDP 201
- Creating diskless devices 204
- Adding a diskless SRDF mirror 205
- Restarting in diskless configuration 206
- Sample session: planned failover 206
- Displaying/querying cascaded SRDF 208

Overview

Cascaded SRDF is a three-way data mirroring and recovery solution that consists of:

- A R1 device replicating data to
- An R21 device at a secondary site, which replicates the same data to a
- R2 device located at a tertiary site

Cascaded SRDF reduces recovery time at the tertiary site because replication continues to the tertiary site if the primary site fails.

This enables a faster recovery at the tertiary site, if that is where the data operation is restarted. You can achieve zero data loss up to the point of the primary site failure.

Figure 27 shows a basic cascaded SRDF configuration.

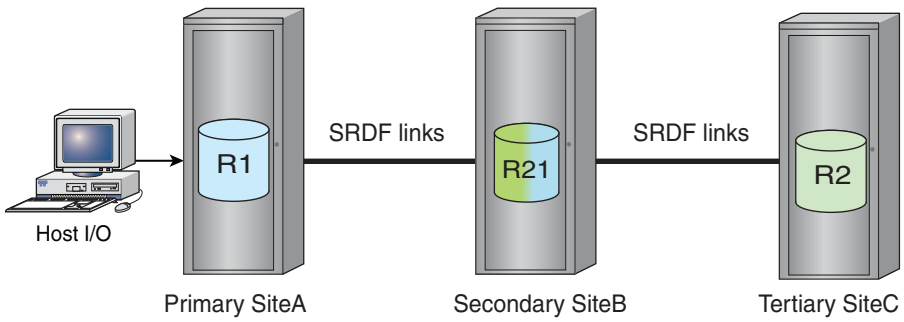


Figure 27 Cascaded SRDF configuration

Note: Cascaded SRDF requires Enginuity 5773 and higher.

Cascaded SRDF uses a new type of SDRF device: the R21 device. An R21 device is both an R1 mirror and an R2 mirror, and is used only in cascaded SRDF configurations.

An R21 device is both:

- An R2 in relation to the R1 source device at the primary site, and
- An R2 in relation to the R2 target device at the tertiary site.

There are two sets of pair states in a cascaded configuration:

- Pair states between the primary and secondary site (R1 → R21)
- Pair states between the secondary and tertiary sites (R21 → R2)

These two pair states are separate from each other.

When performing a control operation on one pair, the state of the other device pair must be known and considered.

"Cascaded SRDF control operations and applicable pair states" on page 383 provides more information.

Note: To perform cascaded SRDF operations with Access Control enabled, you need SRDF BASECTRL, BASE, and BCV access types. *EMC Solutions Enabler Array Management CLI User Guide* provides more information.

SRDF modes in cascaded configurations

The SRDF modes supported on each hop in a cascaded configuration vary depending on whether the R21 device is diskless (EDP is configured).

Table 23 lists the SRDF modes supported from R1 → R21, and R21 → R2 when the R21 device is NOT diskless.

Table 24 lists the SRDF modes supported from R1 → R21, and R21 → R2 when EDP is configured and the R21 device is diskless.

Table 23 SRDF modes for cascaded configurations (no EDP)

R1→R21	R21→R2
Adaptive copy disk	Asynchronous Adaptive copy disk
Adaptive copy write pending*	Asynchronous Adaptive copy disk
Asynchronous (no EDP)	Adaptive copy disk
Semi-synchronous (for 5671 only)	Asynchronous Adaptive copy disk
Synchronous	Asynchronous Adaptive copy disk

* Adaptive Copy Write Pending mode is not supported when the R1 mirror of the RDF pair is on an array running HYPERMAX OS.

Note: Asynchronous mode can be run on either the R1 →› R21 hop, or the R21 → R2 hop, but not both.

SRDF modes in cascaded configurations with EDP

SRDF/Extended Distance Protection (EDP) enables you to designate an R21 device as a diskless device.

A diskless R21 device directly cascades data to the remote R2 disk device, streamlining the linkage and cost of storage at the middle site.

Table 24 SRDF modes for cascaded configurations with EDP

R1→ Diskless R21	Diskless R21→R2
Synchronous Adaptive copy disk Adaptive copy write pending*	Asynchronous
Synchronous Adaptive copy disk Adaptive copy write pending*	Adaptive copy write pending*

*Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS, and diskless R21 devices are not supported on arrays running HYPERMAX OS.

Restrictions (cascaded SRDF)

- The secondary site (R21 devices) must be running Enginuity 5773 or higher.
- R1 and R2 devices must be running Enginuity 5671 or Enginuity 5773 and higher.
- An R21 device cannot be paired with another R21 device

  R1→ R21 → R21 → R2 is not supported.
- R21 devices cannot be BCV devices or PPRC devices.
- R21 devices are supported only on GigE and Fibre RAs.
- If the first device added to an SRDF group is in asynchronous mode (`-rdf_mode async)`, all subsequent devices added to the SRDF group must also be added in asynchronous mode.
- If you do not specify a mode, the option file setting SYMAPI_DEFAULT_RDF_MODE is used. The default is adaptive copy.
- Mixing thin and thick devices is supported with the following restrictions:
  - The arrays that contain the thin devices must be running Enginuity 5876.x.
  - For FBA devices, the arrays that contain the thick devices must be running Enginuity 5876.x.
  - For CKD devices, the arrays that contain the thick devices must be running Enginuity 5876.x.
- If the device to be the R21 device is currently an R1 device, and is in synchronous or adaptive copy write pending mode, creation of the R1 -› R21 relationship is blocked.

  For diskless devices, creation of an R1 device operating in adaptive copy disk is blocked.

  Diskless devices are not supported on arrays running HYPERMAX OS.
- If both SRDF groups for the R21 device are not on a Fibre or GigE director, creation of an R21 device is blocked.
- The same SRDF group cannot be configured for both R21 device mirrors.

Setting up cascaded SRDF

Setting up a cascaded SRDF relationship

Setting up a cascaded SRDF relationship is a two-step process:

1. Create the initial R1 –› R21 pair between array A and array B for the first hop. SRDF/S, SRDF/A, adaptive copy disk mode, or adaptive copy write-pending mode is allowed over the first hop.

   Note: Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.

Note: Only one hop (R1 –› R21 or R21 –› R2) can be asynchronous at a time. If R1 -› R21 is in asynchronous mode, R21 -› R2 must be in adaptive copy disk mode.

2. Create the R21 –› R2 pair between array B and array C for the second hop. SRDF/S, SRDF/A or adaptive copy disk mode is allowed over the second hop.

   The most common implementation is SRDF/S mode for the first hop and SRDF/A mode for the second hop.

Note: For cascaded SRDF without Extended Distance Protection (EDP), the R21 device paired with an R2 device must be in either asynchronous or adaptive copy disk mode

Use the `symrdf createpair` command with the `-rdf_mode` option to create the SRDF pairs for both the first and second hops, and set the SRDF mode.

Note: Use the command twice, once for each hop.

symrdf createpair command syntax (-file option)

```
symrdf -file Filename -sid SymmID -rdfg GrpNum
      [-bypass] [-noprompt] [-i Interval] [-c Count]
      [-v|-noecho] [-force] [-symforce] [-star]

    createpair -type <R1|R2>
      <-invalidate <R1|R2> | -establish | -restore>
      [-rdf_mode <sync|acp_wp|acp_disk|async>]
      [-g NewDg] [-remote]
```

Note: Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.

Example In the following example:

- *TestFile1* specifies two device pairs on SIDs 284 and 305:

```
0380   0390
0381   0391
```

1. Use the `symrdf createpair` command to configure the device pairs, SRDF group, and SRDF mode for the first (R1 -› R2) hop:

```
symrdf createpair -file TestFile1 -sid 305 -rdfg 210 -type R2 -establish -rdf_mode sync
```

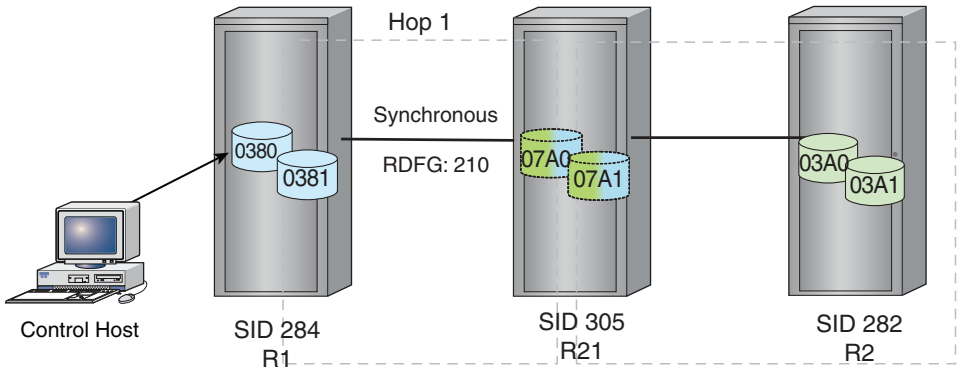


Figure 28 Configuring the first hop

The SRDF R1 -› R2 device pairs are created and established in SRDF synchronous mode.

- *TestFile2* specifies two device pairs on SIDs 305 and 282:
  - 0390 03A0
  - 0391 03A1

2. Use a second `symrdf createpair` command to configure the device pairs, SRDF group, and SRDF mode for the second hop(R21 -› R2):

```
symrdf createpair -file TestFile2 -sid 305 -rdfg 230 -type R1 -establish -rdf_mode acp_disk
```

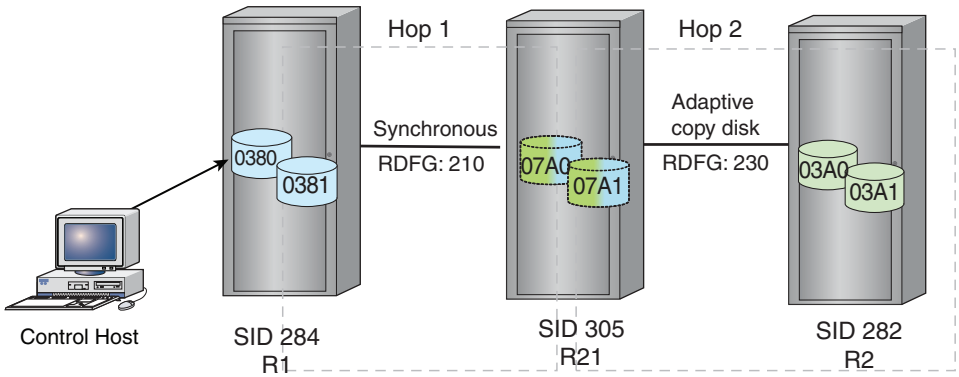


Figure 29 Configuring the second hop

Devices 0390 and 0391 are R21 devices in the cascaded configuration. They are:

- R2 devices in the R1 ->R21 relationship
- R1 devices in the R21-› R2 relationship

Applicable pair states for cascaded SRDF operations

In a cascaded relationship, control operations are only allowed for the pair R1–>R21 when the R21-›R2 pair is in a specific pair state.

"Cascaded SRDF control operations and applicable pair states" on page 383 provides more information.

RDF21 SRDF groups

You can create device groups and composite groups to contain R21 devices as standards. These groups are identified with an SRDF group type: RDF21.

Use the `symdg create` and `symcg create` commands to create device and composite groups with type RDF21.

Examples

To create a device group with SRDF group type RDF21:

```
symdg -type RDF21 create test_group_dg
```

To create a composite group with SRDF group type RDF21:

```
symcg -type RDF21 create test_group_cg
```

To create an RDF1 composite group, add devices and set an SRDF group name:

To create an empty RDF1 composite group `testcg`:

```
symcg -type rdf1 create testcg
```

To add all devices visible to the local host at SID 284 to composite group `testcg`:

```
symcg -cg testcg addall dev -sid 284 -rdfg 210
```

To add all devices visible to the local host at SID 256 to composite group `testcg`:

```
symcg -cg testcg addall dev -sid 256 -rdfg  60
```

To set the SRDF group name to `name1`:

```
symcg -cg testcg set -name name1 -rdfg 284:210,256:60
```

Managing R21 devices

In a cascaded SRDF relationship, the term first hop refers to the R1-› R21 device pair, the term second hop refers to the R21-›R2 device pair.

When controlling an R2 device in a cascaded SRDF relationship, the first hop represents the R2-›R21 relationship and the second hop represents the R21-› R1 relationship.

Operations against one pair relationship depend on the state of the other pair relationship. The SRDF state of the R21 device in a cascaded relationship is determined as follows:

- The SRDF pair state of the R1 -› R21 device is determined by the RA status.
- The SRDF pair state of the R21 -› R2 mirror is determined by the SA status.

Figure 30 illustrates how the R21 SRDF device state is determined and how each SRDF mirrored pair state is determined.

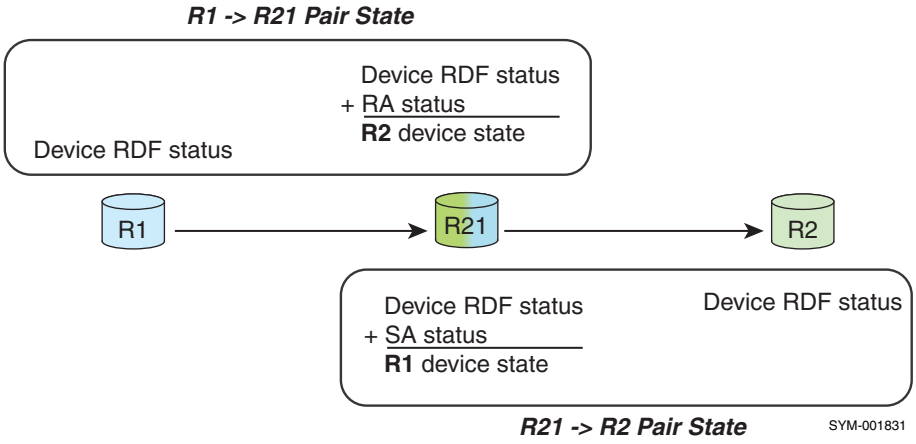


Figure 30 Determining SRDF pair state in cascaded configurations

Device actions modify only the SA status of the R21 device.

For example, if a `rw_enable r1` is performed against the R1 -> R21 pair, and the R21 has a device SA status of `WD`, the overall device SRDF state is `WD`.

You must perform both a `rw_enable r1` against the R21 -> R2 pair and a `rw_enable r2` against the R1 -> R21 pair to make the R21 device `rw_enable` to the host.

Note: If either the R1 or the R2 mirror of an R21 SRDF device is made `NR` or `WD`, the R21 device will be `NR` or `WD` to the host.

"Cascaded SRDF control operations and applicable pair states" on page 383 provides more information.

Hop 2 controls

You can perform control operations from hosts connected any of the three arrays in a cascaded configuration.

Use the `-hop2` option to control an SRDF device that is two hops away. The `-hop2` option can be used with device groups, composite groups, STDs, and local BCVs.

Use the `-hop2` option to control the:

- R21->R2 relationship for an RDF1 device group or composite group
- R1->R21 relationship for an RDF2 device group or composite group

The location of hop-2 devices depends on the location of the controlling host.

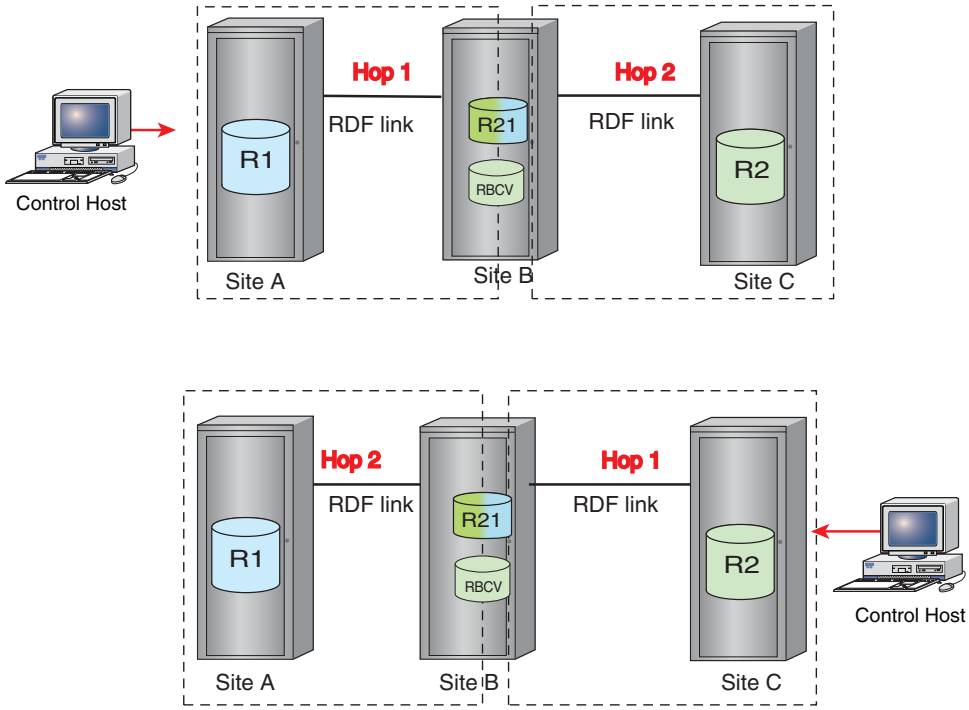


Figure 31 Location of hop-2 devices

In Figure 31:

- When the controlling host is at Site A, a control operation with the `-hop2` option acts on the device pair in the array from Site B to Site C.
- When the controlling host is at Site C, a control operation with the `-hop2` option acts on the device pair in the array from Site B to Site A.

Examples

Use the `-hop2` option with `-rdfg name:` to operate on the second hop SRDF relationship for the specified `-rdfg name:` Assume the following composite group with 4 devices spread across two arrays:

```
CG: testcg     cg type: RDF1   with  R1->R21->R2

Sym: 000192600284  / rdf group  210   /  rdfg name:  name1
R1 device  0380
R1 device  0381


Sym: 000192600256  / rdf group   60   /  rdfg name:  name1
R1 device  0940
R1 device  0941
```

The following command only operates on the R21->R2 SRDF relationships associated with all the R1 devices using SRDF groups named `name1`:

```
symrdf -cg testcg -rdfg name:name1 -hop2 establish
```

Cascaded SRDF with EDP

SRDF/Extended Distance Protection (EDP) streamlines cascaded SRDF linkage to the R2 with a diskless R21 device.

With EDP, replication between the R1 and R2 does not require disks at R21 site.

Note: SRDF/EDP is requires Enginuity 5874 or higher on the array where the R21 device is located. The arrays where the R1 and R2 devices are located require Enginuity 5773 or higher.

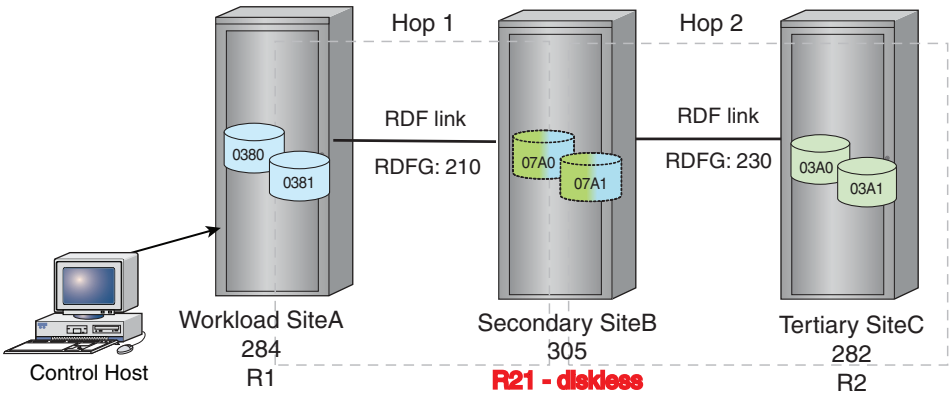


Figure 32 Cascaded SRDF with EDP

Without EDP, the R21 disk device has its own local mirrors so there are three full copies of data, one at each of the three sites.

With EDP, the R21 diskless device has no local mirrors.

Thus, there are only two full copies of data, one on the R1 disk device and one on the R2 disk device.

When using a diskless R21 device, changed tracks received from the R1 mirror are saved in cache until these tracks are sent to the R2 disk device. Once the data is sent to the R2 device and the receipt is acknowledged, the cache slot is freed and the data no longer exists on the R21.

SRDF/EDP rules

The following rules apply when creating diskless SRDF devices:

- For Enginuity 5773, a patch is required to connect to a diskless device.
- A diskless device cannot be mapped to the host. Therefore, no host is able to directly access a diskless device for I/O data (read or write).
- The diskless SRDF devices are only supported on GigE and Fibre RAs.
- Other replication technologies (TimeFinder/Snap, TimeFinder/Clone, Open Replicator, and Federated Live Migration) do not work with diskless devices as the source or the target of the operation.

- The `symreplicate` command returns an error if a diskless device is found in the configuration.
- Diskless devices are not supported with thin CKD devices.
- The R1 and R2 volumes must be *both* thin or *both* standard. For example:
  - Thin R1-› diskless R21-›thin R2, *or*
  - Standard, fully provisioned R1 -› diskless R21 -› standard, fully provisioned R2.

Setting up cascaded SRDF with EDP

Setting up a SRDF/EDP relationship is a two-step process:

1. Create the DLR1 --› R2 pair between array B and array C.
2. Create the R1 --› DLR2 pair between array A and array B.

Now, the configuration is R1 --› DLR21 --› R2.

Table 25 lists the SRDF modes allowed for SRDF/EDP.

Table 25 SRDF modes allowed for SRDF/EDP

R1 - DLR21	DLR21 - R2
Synchronous	Asynchronous
Adaptive copy disk[a]	Asynchronous

a. Adaptive copy mode on the first leg does not provide full time consistency of the R21 or R2 devices.

Use the `symrdf createpair` command with the `-rdf_mode` option to create the SRDF pairs for both the first and second hops, and set the SRDF mode.

Use the command twice, once for each hop.

"symrdf createpair command syntax (-file option)" on page 195 describes the command syntax.

Note: Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.

In an SRDF/EDP configuration, you cannot bring devices Read Write on the link until the diskless devices are designated as being R21s.

Use the `-invalidate R2` option instead of the `-establish` option.

Note: Since the R21 devices are diskless and cannot be mapped, you do not need to make the device Not Ready or Write Disabled before using the `-invalidate R2` option.

Example In the following example procedure, *TestFile1* specifies two device pairs on SIDs 284 and 305:

- 0380 07A0
- 0381 07A1

1. Use the `symrdf createpair` command to configure the device pairs, SRDF group, and SRDF mode for the first (R1 -> R2) hop:

```
symrdf createpair -file TestFile1 -sid 305 -rdfg 210 -type R2 -invalidate R2 -rdf_mode sync
```

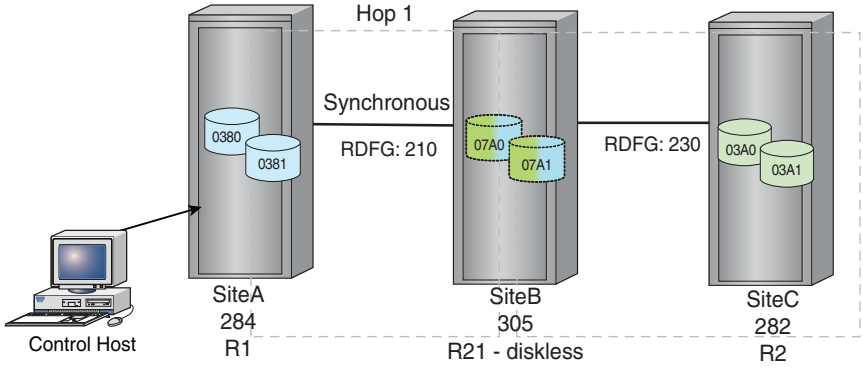


Figure 33 Set up first hop in cascaded SRDF with EDP

The SRDF device pairs are created and placed in synchronous mode.

- *TestFile2* specifies two device pairs:
  - 07A0 03A0
  - 07A1 03A1

2. Use a second `symrdf createpair` command to configure the device pairs, SRDF group, and SRDF mode for the second (R21 -> R2) hop:

```
symrdf createpair -file TestFile3 -sid 305 -rdfg 230 -type R1 -establish -rdf_mode acp_disk
```

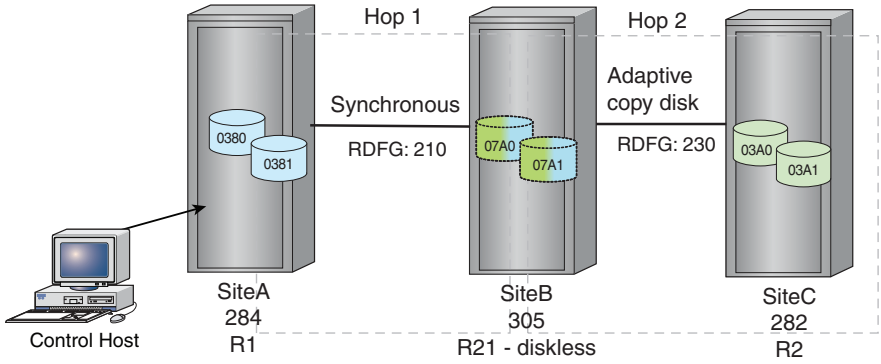


Figure 34 Set up second hop in cascaded SRDF with EDP

3. Use the `symrdf establish` command to make the R1 device pairs Read Write in the first (R1->R21) hop on the link.

```
symrdf establish -file TestFile1 -sid 305 -rdfg 210
```

Rules/restrictions for diskless devices in cascaded SRDF

Note: Diskless devices should only be used as R21 devices in a cascaded environment. Diskless R1, R2, or R22 devices should only be used as an intermediate step to create a diskless R21 device.

General restrictions

- The following control operations are blocked for diskless devices in a R1->R2 relationship that is not part of a cascaded configuration (R1->R2, R2<-->R2, or R1->R22<-R1), or is not going to become part of a cascaded relationship
  - Establish, resume, restore, failback, R1_update, merge
  - Failover if the R2 is a diskless device
  - Createpair –restore or –establish
  - Refresh R1 or swap –refresh R1
  - Refresh R2 or swap –refresh R2
  - Ready/not_ready R1 of a diskless R1 device
  - Ready/not_ready R2 of a diskless R2 device
- A diskless SRDF device may not be paired with another diskless SRDF device.
- For SRDF groups in asynchronous mode, all the devices in the SRDF group must be either diskless or non-diskless.
- You cannot set the skew limit when the R21->R2 hop is in adaptive copy write pending mode. SRDF behaves as if the skew is infinite.
- You must make the link between R21->R2 Ready (RW) before making the R1->R21 link ready (RW). Otherwise, Enginuity makes the diskless R1->R21 devices NR on the link when the R21->R2 state is NR on the link.

Control and set restrictions

You can perform SRDF control and set operations for diskless environments on composite groups, device groups, and files that contain both diskless and non-diskless devices.

Note: You can control SRDF pairs with diskless devices and without diskless devices in a single control operation if some of the R21 devices in the group are diskless and others are not.

- The following configurations are supported when the R21 is a diskless SRDF device:
  - R1->R21->R2
  - R11->R21->R2
  - R11->R21->R22
- You cannot set the mode for an SRDF group containing diskless and non-diskless devices to asynchronous.

  Table 23 on page 193 lists the modes allowed for cascaded SRDF configurations.

Table 24 on page 193 lists the modes allowed for cascaded SRDF configurations where the R21 is diskless.

All other combinations are blocked. If synchronous mode is not allowed, specify a valid SRDF mode when creating these device pairs

⚠CAUTION

The adaptive copy write pending -> asynchronous combination in Table 24 on page 193 cannot reach the Consistent state. The R21->R2 hop hangs in the SyncInProg state with 0 invalid tracks.
To reach the consistent state, configure asynchronous -> asynchronous.

Dynamic control restrictions

You can use dynamic SRDF controls (`createpair`, `deletepair`, `swap_personality`, `movepair`, and `failover -establish` actions) to create and manage diskless device relationships.

The following rules apply for these operations:

- A diskless SRDF device can only be configured on a Fibre or GigE SRDF director.
- A `createpair` action is blocked when both sides are diskless devices.
- The `createpair` and `movepair` actions are blocked if the action results in a mixture of diskless and non-diskless devices in an SRDF group containing devices in asynchronous mode.
- The `createpair`, `movepair`, `swap_personality`, and `failover -establish` actions will be blocked if the action will result in a violation of the allowable SRDF modes as outlined in "Control and set restrictions" on page 203.
- The `createpair` action is blocked if the action results in an R1->R21->R2 relationship where the R1 and the R2 are the diskless devices.

SRDF query restrictions

- A diskless device has no local mirrors. Thus, no local invalid tracks are reported for the device.
- Queries to a diskless R1 device do not show any R1 invalid tracks.
- Queries to a diskless R2 device do not show any R2 invalid tracks.
- Queries to a diskless R21 device do not show any R1 invalid tracks.
- Queries to diskless R21 device do not show any R1 invalid tracks when queried from the R21->R2 relationship point of view.
- Queries to diskless R21 device do not show any R2 invalid tracks when queried from the R1->R21 relationship point of view.

Creating diskless devices

Use the `symconfigure` command to perform control operations (creation, configuration, convert, and delete) for diskless devices, using the following device type designations:

- DLDEV

- RDF1+DLDEV
- RDF2+DLDEV
- RDF21+DLDEV

You can create a diskless device using the existing `create/configure dev` command with one of the these device types.

You cannot create an RDF21+DLDEV device directly. Use the `add rdf mirror` command with `symconfigure` to create R21 diskless devices. "Adding a diskless SRDF mirror" on page 205 provides more information.

You can also use the `set dev` command with `symconfigure` to set attributes on diskless devices.

Note: For more information about the `symconfigure`, see the *EMC Solutions Enabler Array Management CLI User Guide*.

Adding a diskless SRDF mirror

Use the `symconfigure` `add rdf mirror` command to add both static and dynamic SRDF mirrors to diskless devices.

The procedure to set up a diskless R21 device is the same as any other type of R21 device.

In order to add the diskless device, it must already be an RDF1+DLDEV or an RDF2+DLDEV device:

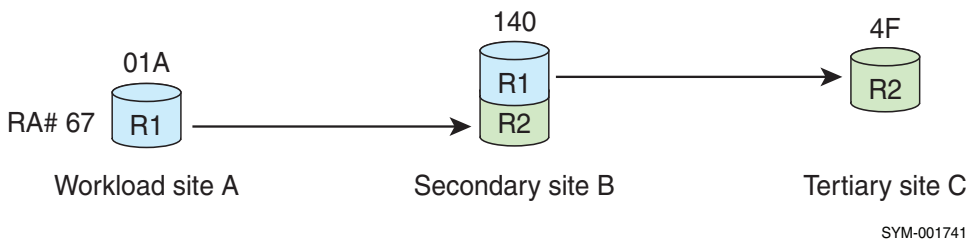


Figure 35 Adding a diskless SRDF mirror

Use the `symconfigure` command to add the R21 mirrors.

Perform the `add rdf mirror` command twice; once for each site.

Example To add the specified device from site A:

```
add rdf mirror to dev 01A
  ra_group=67, mirror_type=RDF1
  remote_dev=140
...
```

To add the specified device from site C:

```
add rdf mirror to dev 04F
  ra_group=67, mirror_type=RDF2
  remote_dev=140
```

Restrictions

- Either the local or the remote device can be diskless, however, both the local and the remote SRDF device cannot be diskless.

- Diskless devices can only be configured on a fibre or GigE SRDF directors.
- Cannot add a mix of diskless and non-diskless SRDF devices to an SRDF group with devices in Async mode.
- The create pair action is blocked if it results in an R1->R21->R2 relationship where the R1 and the R2 are diskless devices.
- When configuring a diskless device the modes should be set as per rules discussed in "Control and set restrictions" on page 203.

Restarting in diskless configuration

- When restarting a diskless SRDF configuration:

  The R21->R2 hop is recovered before the R1->R21 hop.

  The R1->R21 relationship cannot be RW on the link when the R21->R2 relationship is NR on the link.

- When recovering with a diskless R21 device:

  The `restart_sync_type` is in adaptive copy write pending mode for the R21->R2 relationship.

  Adaptive copy write pending mode (acp_wp) is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS, and diskless R21 devices are not supported on arrays running HYPERMAX OS.

Sample session: planned failover

This section is an example of a planned failover of the cascaded SRDF configuration depicted in Figure 36:

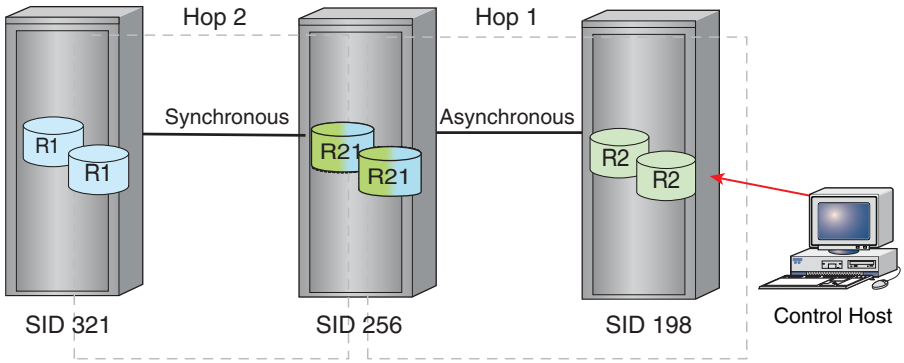


Figure 36 Cascaded configuration before planned failover

For the example session:

- Commands are issued from a control host connected to SID 198.
- Commands are issued to an SRDF device group.

1. Use the **`symcfg list`** command to verify that both array 321 and 256 are visible to the control host.

2. Use the **symrdf -g *GroupName* query -hop2** command to verify that the RDF Pair State for devices in the SID 321 -> SID 256 hop are Synchronized.

   The SID 321 -> SID 258 hop is synchronous. Healthy device pairs are “Synchronized”.

3. Use the **symrdf -g *GroupName* query -rdfa** command to verify that the RDF Pair State for devices in the SID 256 -> SID 198 hop are Consistent.

   The SID 256 -> SID 198 hop is asynchronous. Healthy device pairs are “Consistent”.

4. Use the **symrdf -g *GroupName* suspend -hop2** command to suspend the device pairs of the SID 321 -> SID 256 hop.
5. Use the **symrdf -g *GroupName* query -hop2** command to verify that the RDF Pair State for devices in the SID 321 -> SID 256 hop is Suspended.
6. Use the **symrdf -g *GroupName* suspend -force** command to suspend the device pairs of the SID 256 -> SID 198 hop.
7. Use the **symrdf -g *GroupName* query** command to verify that the RDF Pair State for devices in the SID 256 -> SID 198 hop is Suspended.
8. Use the **symrdf -g *GroupName* failover -hop2** command to failover from SID 321 to SID 256.
9. Use the **symrdf -g *GroupName* failover -force** command to failover from SID 256 to the SID 198.
10. Use the **symrdf -g *GroupName* query -hop2** command to verify that the RDF Pair State for devices in the SID 321 -> SID 256 hop are Failed Over.
11. Use the **symrdf -g *GroupName* query** command to verify that the RDF Pair State for devices in the SID 256 -> SID 198 hop are Failed Over.
12. Use the **symrdf -g *GroupName* set mode acp_disk -hop2** command to change the SRDF mode between SID 321 and SID 256 to adaptive copy disk mode.
13. Use the **symrdf -g *GroupName* swap -hop2** command to swap personalities between SID 321 and SID 256.

    The configuration is now:

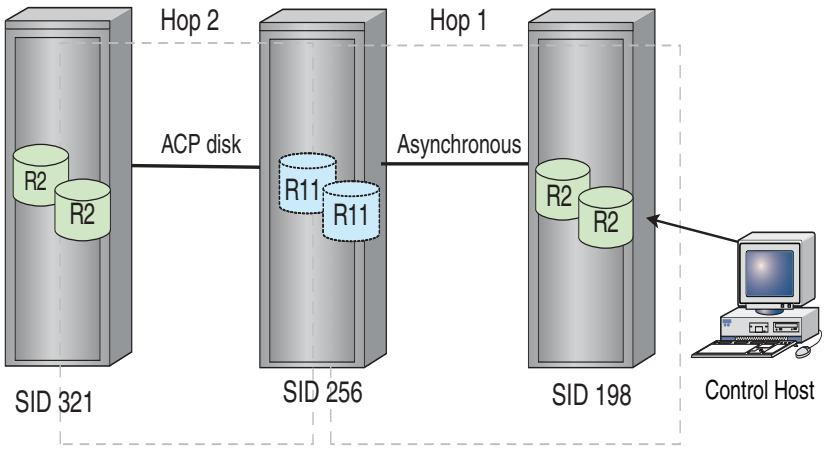


Figure 37 Planned failover - after first swap

14. Use the **symrdf -g *GroupName* swap** command to swap personalities between SID 256 and SID 198.

    The configuration is now:

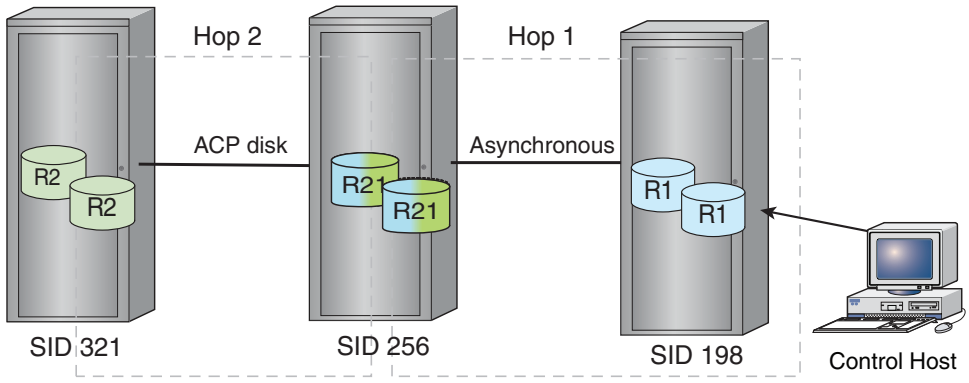


Figure 38 Planned failover - after second swap

15. Use the **symrdf -g *GroupName* resume -hop2** command to resume the device pairs of the SID 256 -> SID 321 hop.
16. Use the **symrdf -g *GroupName* resume -force** command to resume the device pairs of the SID 198 -> SID 256 hop.

Note: Do not change the SRDF mode from SID 256 -> SID 321. The R1 -> R21 hop is now Asynchronous. Only adaptive copy disk mode is supported for the R21 -> R2 hop.

Displaying/querying cascaded SRDF

See alsoTable 6, “Options for symrdf list command”and Table 5, “Commands to display and verify SRDF, devices, and groups”.

Listing cascaded SRDF devices

- Use the `symrdf list` command with the `-R21` option to displays all R21 devices.

  This option cannot be specified in the same command with the `-R1` or `-R2` option.
- Use the `-cascade` option to list all R21 devices and the R1 and R2 devices with which they are paired.

  This option also lists R1 and R2 devices participating in cascaded SRDF relationships.
- Use the `-cascade` option in conjunction with the `-R1`, `-R2`, or `-R21` options to display only R1, R2, or R21 devices participating in cascaded SRDF relationships.

  R21 devices and the devices with which they are paired are considered concurrent devices. You can also use the `-concurrent` flag to display these devices.

Listing R21 devices

Output of the `symrdf list` command includes the SRDF Mirror Type associated with the SRDF group.

In the following example, Mirror Type is in bold text.

```
symrdf list -sid 305 -cascaded

Symmetrix ID: 000192600305

                              Local Device View
---------------------------------------------------------------------------
                       STATUS     MODES                      RDF  S T A T E S
Sym          RDF       ---------  -----  R1 Inv   R2 Inv ----------------------
Dev    RDev  Typ:G     SA RA LNK  MDATE  Tracks   Tracks Dev RDev Pair
----   ---- --------   ---------  ----- -------  ------- --- ---- -------------

00390 00380  R21:210   RW WD RW   S..2.       0        0 WD  RW   Synchronized
      003A0  R21:230   RW RW RW   C.D1.       0        0 RW  WD   Synchronized
00391 00381  R21:210   RW WD RW   S..2.       0        0 WD  RW   Synchronized
      003A1  R21:230   RW RW RW   C.D1.       0        0 RW  WD   Synchronized
.
.
.
Legend for MODES:

 M(ode of Operation)    : A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)               : X = Enabled, . = Disabled
 A(daptive Copy)        : D = Disk Mode, W = WP Mode, . = ACp off
 (Mirror) T(ype)        : 1 = R1, 2 = R2
 (Consistency) E(xempt): X = Enabled, . = Disabled, M = Mixed, - = N/A
```

Viewing diskless devices

IMPORTANT

`symcg`, `symdg`, or `symdev` commands used with `relabel` option fail when the scope includes any diskless device.

symrdf list

Use the `symrdf list` command with the `-diskless_rdf` option to view SRDF diskless devices.

You can use the `-diskless_rdf` option with `-R1`, `-R2`, `-R21`, or `-dynamic`.

The output displays the requested diskless SRDF or SRDF capable devices.

Example To display SRDF diskless devices:

```
symrdf list -diskless_rdf
```

symdev list

Use the `symdev list` command with the `-dldev` option to display all configured diskless devices.

You can use the `-dldev` option with `-R1`, `-R2`, `-R21`, or `-dynamic`.

Example To display all diskless devices for Symm 305:

```
symdev list -sid 305 -dldev
```

```
Symmetrix ID: 000192600305

        Device Name           Directors                   Device
----------------------------  -------------  -------------------------------------
                                                                                Cap
Attribute     Sts    (MB)
----------------------------  -------------  -------------------------------------

007A0 Not Visible             ???:? ???:?   RDF21+DLDEV    Grp'd      RW    1031
007A1 Not Visible             ???:? ???:?   RDF21+DLDEV    Grp'd      RW    1031
```

symdev show

The `symdev show` command displays the following information related to diskless devices:

- `Device Configuration` - shows the device as being an R21 diskless device.
- `Device SA Status` - always N/A. Diskless devices cannot be mapped to a host.
- `Paired with Diskless Device` - indicates if the device is in an SRDF relationship with a diskless SRDF device, and the device type for the SRDF partner of this device.

Example To display information about the specified diskless device:

```
symdev show 07A0 -sid 05
.
    Device Configuration      : RDF21+DLDEV      (Non-Exclusive Access)
.
.
Device Status             : Ready            (RW)

    Device SA Status          : N/A              (N/A)

    Mirror Set Type           : [R2 Remote,R1 Remote,N/A,N/A]

    Mirror Set DA Status      : [RW,RW,N/A,N/A]

    Mirror Set Inv. Tracks    : [0,0,0,0]

    Back End Disk Director Information
        {
        Hyper Type                              : R2 Remote
        Hyper Status                            : Ready            (RW)
        Disk [Director, Interface, TID]         : [N/A,N/A,N/A]
        Disk Director Volume Number             : N/A
        Hyper Number                            : N/A
        Mirror Number                           : 1

        Hyper Type                              : R1 Remote
        Hyper Status                            : Ready            (RW)
        Disk [Director, Interface, TID]         : [N/A,N/A,N/A]
        Disk Director Volume Number             : N/A
        Hyper Number                            : N/A
        Mirror Number                           : 2
...
}
    RDF Information
        {
        Device Symmetrix Name                   : 007A0
        RDF Type                                : R2
        RDF (RA) Group Num                      : 210              (D1)

        Remote Device Symmetrix Name            : 00380
```

```
                Remote Symmetrix ID                    : 000192600284

                R2 Device Is Larger Than The R1 Device : False
                Paired with Diskless Device            : False
                Concurrent RDF Relationship            : False
                Cascaded RDF Relationship              : True
        ...
            RDF Information
                {
                Device Symmetrix Name                  : 007A0
                RDF Type                               : R1
                RDF (RA) Group Num                     : 230             (E5)

                Remote Device Symmetrix Name           : 003A0
                Remote Symmetrix ID                    : 000192600282

                R2 Device Is Larger Than The R1 Device : False
                Paired with Diskless Device            : False
                Paired with a Concurrent RDF Device    : False
                Paired with a Cascaded RDF Device      : False
        ...
```

Querying hop 2 information

Use the `symrdf -cg CGName -rdfg name:name -hop2 query` command to display information about the second hop SRDF pair of a cascaded SRDF relationship, for the specified subset of the composite group.

Example To display second hop information for composite group `testcg`:

```
symrdf -cg testcg -rdfg name:name1 -hop2 query

Composite Group Name      :  testcg
Composite Group Type      :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  NONE


Symmetrix ID                      : 000192600284    (Microcode Version: 5874)
Hop-2 Symmetrix ID                : 000192600305    (Microcode Version: 5874)
Hop-2 Remote Symmetrix ID         : 000192600282    (Microcode Version: 5874)
RDF (RA) Group Number             : 210 (D1)
Hop-2 RDF (RA) Group Number       : 230 (E5)

        Source (R1) View             Target (R2) View       MODES STATES
--------------------------------  ------------------------- ----- ------ ------------
               ST                 LI       ST                      C  S
  Standard     A                  N        A                       o  u
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Device   Dev   E  Tracks Tracks   S  Dev   E  Tracks  Tracks MDAE  s  p   STATE
--------------------------------  -- ------------------------- -----  ------ ------------
DEV001   00390 RW       0       0 RW 003A0 WD       0       0 C.D.  .  -   Synchronized
DEV002   00391 RW       0       0 RW 003A1 WD       0       0 C.D.  .  -   Synchronized

Symmetrix ID                      : 000192600256    (Microcode Version: 5874)
Hop-2 Symmetrix ID                : 000192600321    (Microcode Version: 5874)
Hop-2 Remote Symmetrix ID         : 000192600198    (Microcode Version: 5874)
RDF (RA) Group Number             :  60 (3B)
Hop-2 RDF (RA) Group Number       :  70 (45)

        Source (R1) View             Target (R2) View       MODES STATES
--------------------------------  ------------------------- ----- ------ ------------
              ST                 LI       ST                      C  S
  Standard     A                  N        A                       o  u
```

```
Logical  Sym   T   R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Device   Dev   E   Tracks  Tracks   S Dev   E  Tracks  Tracks MDAE  s  p   STATE
-------------------------------- -- ------------------------ ----- ------ ------------
DEV003   00944 RW       0       0 RW 00942 WD       0       0 C.D.  .  -   Synchronized
DEV004   00945 RW       0       0 RW 00943 WD       0       0 C.D.  .  -   Synchronized

Total            ------- -------            ------- -------
  Track(s)             0       0                  0       0
  MBs                0.0     0.0                0.0     0.0

Legend for MODES:

 M(ode of Operation)   : A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)              : X = Enabled, . = Disabled
 A(daptive Copy)       : D = Disk Mode, W = WP Mode, . = ACp off
 (Consistency) E(xempt): X = Enabled, . = Disabled, M = Mixed, - = N/A

Legend for STATES:

 Cons(istency State)   : X = Enabled, . = Disabled, M = Mixed, - = N/A
 Susp(end State)       : X = Online, . = Offline, P = Offline Pending, - = N/A
```

Query output information

- Number of SRDF (RA) Groups — Represents the number of R1 -> R21 SRDF groups in the composite group.
- Symmetrix ID — Symmetrix ID of the R1 device.
- Hop-2 Symmetrix ID — Symmetrix ID of the R21 device.
- Hop-2 Remote Symmetrix ID — Symmetrix ID of the R2 device.
- SRDF (RA) Group Number — SRDF group of the R1 device.
- Hop-2 SRDF (RA) Group Number — SRDF group of the R21 device.
- Total — Sums the invalid tracks (and MB) across all displayed R21 -> R2 SRDF groups (that is, it sums all hop-2 invalid tracks).

Note: With an R1->R21-> R2 configuration, issuing a `query -hop2` from an RDF1 composite group indicates that the query should show the relationship of the R21-> R2 device pairs. Thus the query displays the R21 device from the R1 mirror point of view (and vice versa for RDF2 CG).

To see both hops of the RDF1 or RDF2 CG that contains devices in a cascaded SRDF relationship, use the `symrdf -cg query` command with the `-hop2` and the `-detail` options.

Detailed output

To display detailed information about the second hop SRDF pair of a cascaded SRDF relationship, use the `-detail` option with the `symrdf query` command.

Detailed output displays the association of the cascaded pair with the appropriate local pair.

Note: The `-detail` option is not supported for a device group.

Example To display detailed information about the second hop SRDF pair of a cascaded SRDF relationship for composite group `testcg`:

```
symrdf query -cg testcg -rdfg name:name1 -hop2 -detail

Composite Group Name      :  testcg
Composite Group Type      :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  NONE

RDFG Names:
    {
    RDFG Name                          : name1
    RDF Consistency Mode               : NONE
    }

Symmetrix ID                      : 000192600284    (Microcode Version: 5874)
Remote Symmetrix ID               : 000192600305    (Microcode Version: 5874)
RDF (RA) Group Number             : 210 (D1) - name1

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ------------
              ST                 LI       ST
  Standard     A                  N        A
Logical  Sym   T  R1 Inv  R2 Inv  K        T   R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev    E   Tracks  Tracks MDACE STATE
-------------------------------- -- ------------------------- ----- ------------
DEV001   00380 RW       0       0 RW 00390 WD       0       0 S.... Synchronized
DEV002   00381 RW       0       0 RW 00391 WD       0       0 S.... Synchronized

Hop-2
    {
    Symmetrix ID                      : 000192600305    (Microcode Version: 5874)
    Remote Symmetrix ID               : 000192600282    (Microcode Version: 5874)
    RDF (RA) Group Number             : 230 (E5)

            Source (R1) View             Target (R2) View       MODES
    --------------------------------  ------------------------- ----- ------------
                  ST                 LI       ST
      Standard     A                  N        A
    Logical  Sym   T  R1 Inv  R2 Inv  K        T   R1 Inv  R2 Inv       RDF Pair
    Device   Dev   E  Tracks  Tracks  S Dev    E   Tracks  Tracks MDACE STATE
    -------------------------------- -- ------------------------- ----- ------------
    DEV001   00390 RW       0       0 RW 003A0 WD       0       0 C.D.. Synchronized
    DEV002   00391 RW       0       0 RW 003A1 WD       0       0 C.D.. Synchronized
    }

Symmetrix ID                      : 000192600256    (Microcode Version: 5874)
Remote Symmetrix ID               : 000192600321    (Microcode Version: 5874)
RDF (RA) Group Number             :  60 (3B) - name1

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ------------
              ST                 LI       ST
  Standard     A                  N        A
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDACE STATE
--------------------------------- -- ------------------------- ----- ------------
DEV003   00940 RW       0       0 RW 00944 WD       0       0 S.... Synchronized
DEV004   00941 RW       0       0 RW 00945 WD       0       0 S.... Synchronized

Hop-2
{
    Symmetrix ID                      : 000192600321    (Microcode Version: 5874)
    Remote Symmetrix ID               : 000192600198    (Microcode Version: 5874)
```

```
    RDF (RA) Group Number             :  70 (45)

            Source (R1) View             Target (R2) View       MODES
    --------------------------------  ------------------------- ----- ------------
                   ST                 LI       ST
      Standard      A                  N        A
    Logical  Sym    T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
    Device   Dev    E  Tracks Tracks   S Dev    E  Tracks  Tracks MDACE STATE
    -------------------------------- -- ------------------------- ----- ------------
    DEV003   00944 RW       0       0 RW 00942 WD       0       0 C.D.. Synchronized
    DEV004   00945 RW       0       0 RW 00943 WD       0       0 C.D.. Synchronized
    }

Total            ------- -------            ------- -------
  Track(s)             0       0                  0       0
  MBs                0.0     0.0                0.0     0.0

  Hop-2 Track(s)       0       0                  0       0
  Hop-2 MBs          0.0     0.0                0.0     0.0

Legend for MODES:

 M(ode of Operation)   : A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)              : X = Enabled, . = Disabled
 A(daptive Copy)       : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State)   : X = Enabled, . = Disabled, M = Mixed, - = N/A
 (Consistency) E(xempt): X = Enabled, . = Disabled, M = Mixed, - = N/A
```

Query output information

- Symmetrix ID — Symmetrix ID of the R1 device if outside a Hop-2 {. . .} group, or the Symmetrix ID of the R21 device if inside a Hop-2 {. . .} group.
- Remote Symmetrix ID — Symmetrix ID of the R21 device if outside a Hop-2 {. . .} group, or the Symmetrix ID of the R2 device if inside a Hop-2 {. . .} group; had this been an RDF2 CG, then Remote Symmetrix ID inside a Hop-2 {. . .} group would represent the Symmetrix ID of the R1 device.
- SRDF (RA) Group Number — SRDF group from the R1->R21 devices if outside a Hop-2 {. . .} group, or the SRDF group from the R21->R2 devices if inside a Hop-2 {. . .} group; had this been an RDF2 CG, then SRDF (RA) Group Number inside a Hop-2 {. . .} group would represent the SRDF group from the R21->R1 devices.

Note: As with the `-hop2 query` of the same CG illustrated earlier, different R21->R2 SRDF groups are reported separately.

CHAPTER 8
SRDF/Star Operations

This chapter describes the following topics.

- Overview 216
- Understanding SRDF/Star states and operations 221
- SRDF/Star operations - summary 232
- Configure and bring up SRDF/Star 236
- Basic SRDF/Star operations 251
- SRDF/Star consistency group operations 253
- Recovery and switch operations: concurrent SRDF/Star 264
- Recovery and switch operations: cascaded SRDF/Star 282
- Reconfiguration operations 296
- SRDF/Star configuration with R22 devices 304

Overview

SRDF/Star deployments include three geographically dispersed data centers. SRDF/Star protects against a primary site failure or a regional disaster by mirroring production data synchronously to a nearby site and asynchronously to a distant site. This architecture can be expanded to include multiple triangles.

If a failure occurs at the workload site, one target site resumes data replication for the workload site while the other resumes as a protected secondary target site.

SRDF/Star uses dynamic SRDF devices that can function as either an R1 or an R2 device. During failure recovery, the R2 devices at either the synchronous target site or the asynchronous target site are dynamically converted to R1 devices to become production devices at the new workload site.

The basic component of the SRDF/Star configuration is the composite group (CG). Multi-session Consistency (MSC) or Enginuity Consistency Assist (ECA) technology ensures data consistency, and that all members in the CG are either replicating or not replicating.

The CG definition can span cascaded and concurrent SRDF configurations (SRDF/A and SRDF/S) across multiple arrays.

Note: SRDF/Star requires a control host at the workload site, SRDF/A recovery links, and a control host at one of the target sites.

SRDF/Star topologies include

- "Concurrent SRDF/Star", and
- "Cascaded SRDF/Star"

SRDF/Star topologies can also include concurrent devices, as described in:

- "SRDF/Star concurrent R22"

Concurrent SRDF/Star

In a concurrent configuration, data at the workload site is replicated directly to two remote target sites:

- The "synchronous target site" is within synchronous distances and is linked to the workload site by SRDF/S replication.
- The "asynchronous target site" can be hundreds of miles from the workload site and is linked to the workload site by SRDF/A replication.

Concurrent SRDF/Star environments dramatically reduce the time to reestablish replication operations in the event of a failure.

Figure 39 shows a concurrent SRDF/Star configuration:

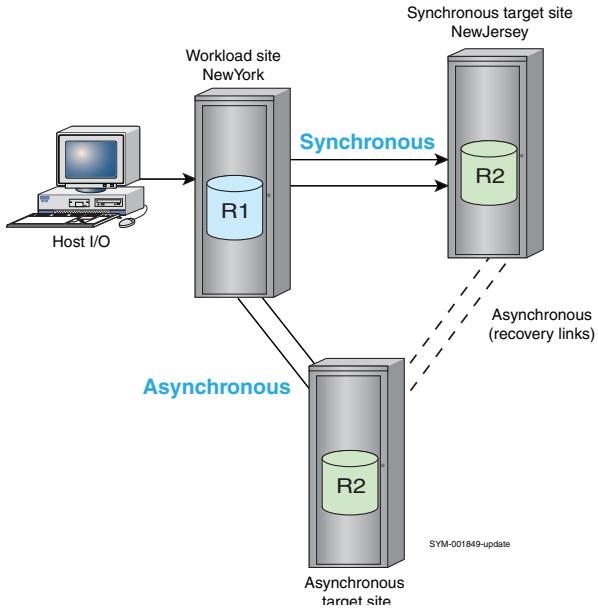


Figure 39 Concurrent SRDF/Star configuration

Data transfer from the workload site is:

- Synchronous to the nearby target site (NewYork) and,
- Asynchronous to the distant target site (London).

During normal operations, the recovery links between synchronous target site and the asynchronous target site are inactive.

In the event of an outage at the workload site, an SRDF/A session can be quickly established between the two target sites.

In the event of a rolling disaster at the workload site, you can determine which target site contains the most current data.

Cascaded SRDF/Star

In a cascaded configuration, data at the workload site is replicated to a “synchronous target site” within synchronous distances.

The data is then replicated from the “synchronous target site” to a more remote “asynchronous target site”.

Cascaded SRDF/Star environments also dramatically reduce the time to reestablish replication operations.

Figure 40 shows a cascaded SRDF/Star configuration:

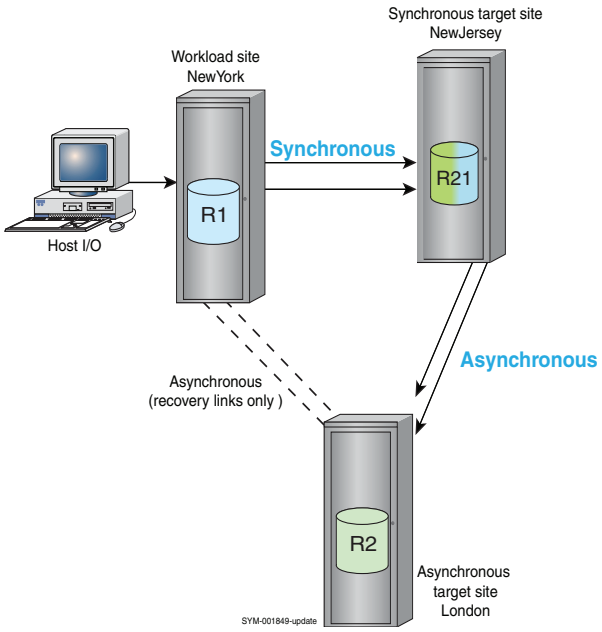


Figure 40 Cascaded SRDF/Star configuration

In cascaded SRDF/Star, the synchronous target site is always more current than the asynchronous target site, but you can choose which site's data to use for recovery.

SRDF/Star concurrent R22

Note: Concurrent devices require Enginuity 5773.150 and higher.

R22 devices (concurrent R2 devices) are specifically designed for SRDF/Star configurations to simplify failover and improve the resiliency of SRDF/Star applications. R22 devices significantly reduce the number of steps needed for `reconfigure`, `switch`, and `connect` commands.

Figure 41 on page 219 shows a typical concurrent SRDF/Star using R22 devices:

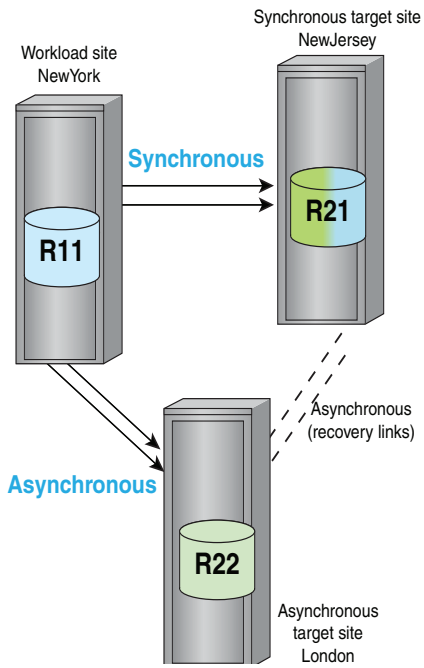


Figure 41 Typical concurrent SRDF/Star with R22 devices

Figure 42, shows a typical cascaded SRDF/Star using R22 devices:

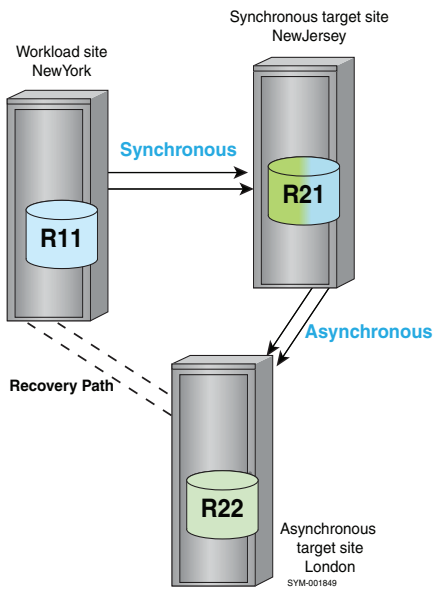


Figure 42 Typical cascaded SRDF/Star with R22 devices

R11and R22 devices have two mirrors, each paired with a different mirror.

Only one of the R22 mirrors can be active (read/write) on the link at a time.

SRDF/Star features

- Differential synchronization greatly reduces the time to establish remote mirroring and consistency.

- In the event of a workload site failure occurs, SRDF/Star reduces the time to failover and resume asynchronous data transfer between the remaining target sites.
- In the event of a rolling disaster at the workload site, you can choose which of the target sites holds the more current data and switch workload operations to that site.
- You can add and remove devices from an SRDF consistency group to maintain data consistency without interrupting the workload.

Restrictions

- If any array in a SRDF/Star configuration is running HYPERMAX OS, Solutions Enabler 8.0.1 or higher is required in order to issue Star controls.
- Redundant control hosts at the workload site and a target site are required.
- Each control host must be connected to only one site in the SRDF/Star triangle.

  EMC strongly recommends running redundant SRDF daemons on multiple control hosts to ensure that at least one SRDF daemon is available to perform time-critical, consistency monitoring operations. Redundant SRDF daemons avoid service interruptions caused by performance bottlenecks local to a control host.
- An SRDF daemon must be running on at least one control host attached locally to each site.
- SRDF/A recovery links are required.
- SRDF groups cannot be shared between separate SRDF/Star configurations.
- R22 devices are required in SRDF/Star environments that include VMAX 10K or VMAXe arrays.
- CKD striped metadevices are not supported.
- R2 devices larger than their R1 devices are not supported.
- Composite groups consisting of device groups are not supported.
- Devices cannot be BCV devices.
- Every device must be dynamic SRDF (R1 and R2 capable).
- BCV device management must be configured separately.

  Note: EMC strongly recommends that you have BCV device management available at both the synchronous and asynchronous target sites.

- Enginuity 5874.228.182 and higher supports concurrent and cascaded SRDF/Star environments with thin devices.
- With Enginuity 5876.159.102 and higher, a mixture of thin and (non-diskless) thick devices is supported.

  Note: If the thick device is on a DMX array running Enginuity 5773.184.130 and higher, thick-to-thin migration is supported if the VMAX array is running Enginuity 5876.163.105 and higher.

Understanding SRDF/Star states and operations

The state of your SRDF/Star environment determines which operations you can perform, and includes:

- The "SRDF/Star state"of your configuration,
- "Target site states", and
- The location of the workload site and target sites.

SRDF/Star state

SRDF/Star state refers to the workload site and both target sites as a complete entity.

Table 26 describes the SRDF/Star states.

Table 26 SRDF/Star states

State	Description
Star Protected	There is data flow and consistency protection at each target site. SDDF sessions are tracking the differences between the sites. If the workload site failed, a differential synchronization between the two target sites would be possible.
Star Tripped	There is no data flow between the workload site and at least one of the target sites.
Star Unprotected	A differential synchronization between the target sites would not be possible.

IMPORTANT

Your configuration must be in the Star Protected state to in order to have SRDF/Star's consistent data protection and incremental recovery capabilities.

Target site states

SRDF/Star target site state refers to the relationship between the target sites and the workload site.

Table 27 describes the SRDF/Star target site states.

Table 27 SRDF/Star target site states

State	Description
Disconnected	May indicate that there is no data flow between the workload site and the target sites. **Note:** If SRDF/Star cannot determine the site state, it will report the state as Disconnected even though there may still be data flow between the sites.
Connected	There is data flow between the sites. The target site is not necessarily synchronized with the workload site.
Protected	There is data flow between the sites. Dependent write consistency of the data at the target site is assured.

Table 27 SRDF/Star target site states

State	Description
Halted	There is no data flow between the sites. There is no data protection at the target site relative to the workload site. The data at each site is the same.
Isolated	There is no data flow between the sites. The devices at the target site are read/write enabled to their local host.
PathFail	There is no data flow between the sites. **Note:** Occurs only if the specified target was in a Protected state. The PathFail;CleanReq state indicates that the `cleanup` operation is required to perform MSC cleanup on the async target before it will be consistent.

SRDF/Star site transitions

In the following discussion, the initial configuration of is as follows:

- Site A is the workload site,
- Site B is the nearby synchronous target site, and
- Site C is the distant asynchronous target site.

After a switch or reconfiguration, the workload site can shift to sites B or C.

The new location of the synchronous target and the asynchronous target varies based on the new configuration.

In cascaded configurations, there are two possible configurations when the workload is at site C:

- Site A is the first hop toward Site B
- Site B is the first hop toward Site A

The following diagrams show the site locations for the original configuration (workload is at Site A), and after the workload has been moved to Site B or C.

NOTICE

When the workload is at site C:

- Both of the target sites are long-distance links, so neither site can be synchronously mirrored.
- It can only be protected at one other site and it can never become fully STAR protected.

In the following diagrams, one of the targets is labeled as the (Sync) target in order to differentiate between the two target sites.

Figure 43 shows site configurations for Cascaded SRDF/Star and Concurrent SRDF/Star without concurrent devices.

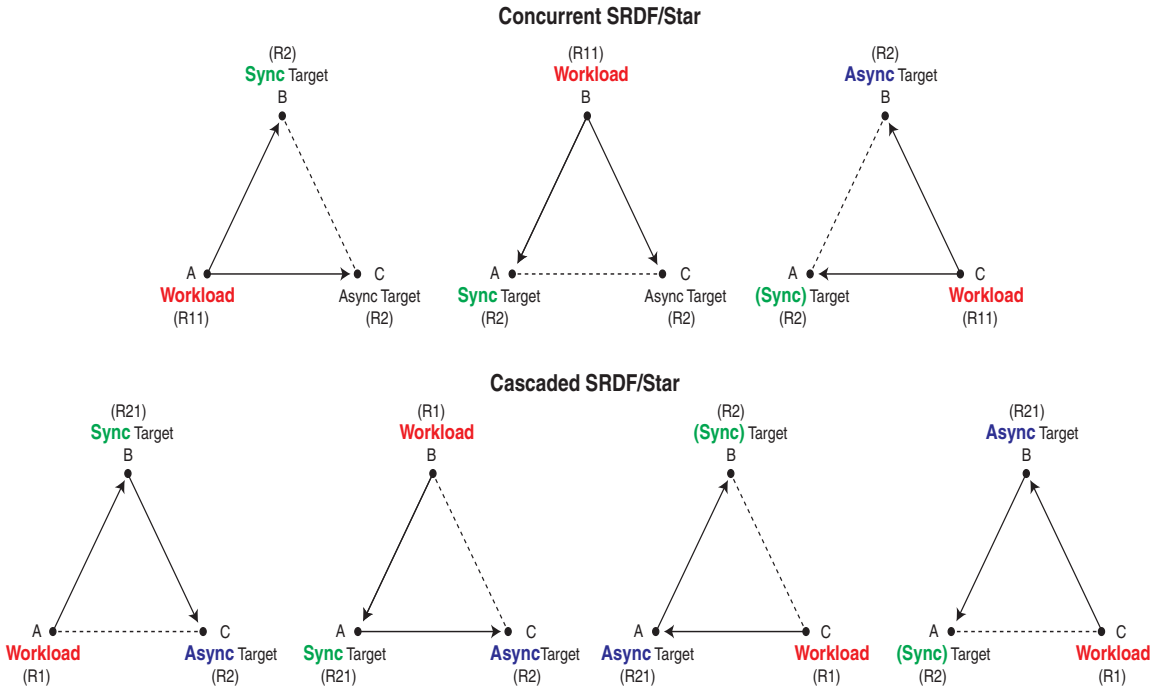


Figure 43 Site configuration transitions without concurrent devices

Figure 44 shows the site configurations for Cascaded SRDF/Star and Concurrent SRDF/Star with concurrent devices.

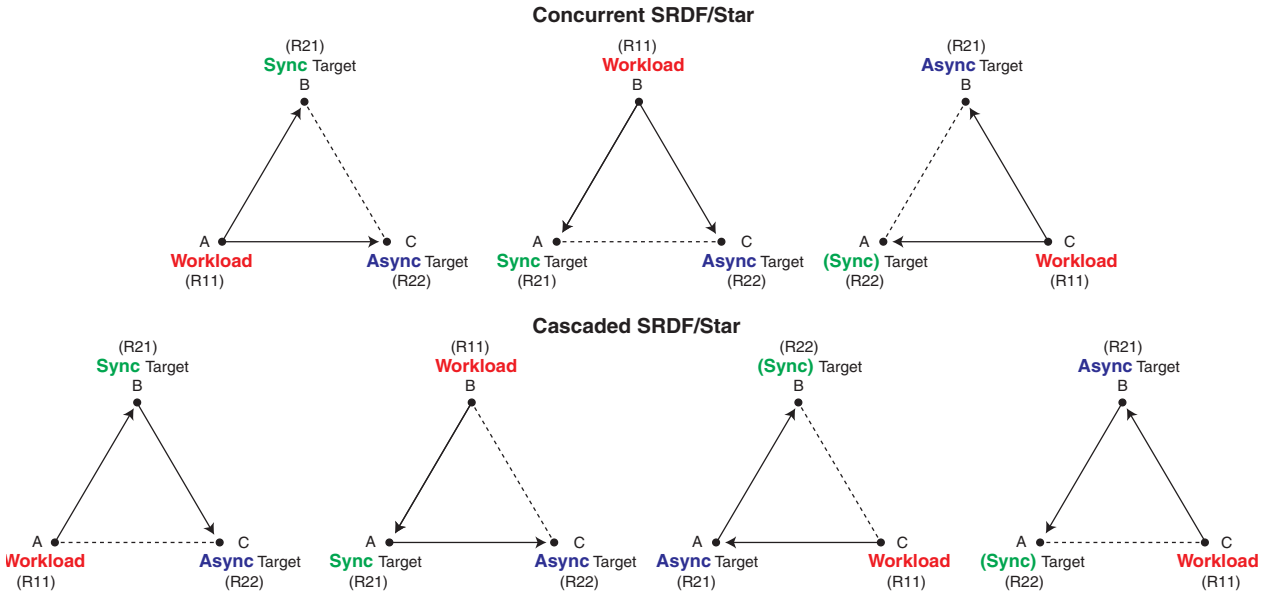


Figure 44 Site configuration transitions with concurrent devices

Operation categories

SRDF/Star operations can be broken into four categories. Table 28 describes the types of SRDF/Star operations.

Table 28 SRDF/Star operation categories

Operation Category	Description
Normal operations	Used to configure and setup SRDF/Star to achieve SRDF/Star protection. Includes the actions required to isolate a site for testing or other required data processing.
Transient fault operations	Used to recover from a temporary failure caused by loss of network connectivity or either target site. Transient faults do not disrupt production at the workload site, so these operations can be executed at the workload site.
Switch operations	**Planned**: Used to move the production workload to a new site with a planned procedure. Planned switch operations are often used for maintenance purposes. They can also be used to return the workload to the original workload site after a disaster forced a move of production activity to one of the target sites. **Unplanned**: Used to recover from faults caused by the loss of a workload site. The loss of a workload site requires an unplanned switch of the workload to one of the target sites.
Reconfigure operations	**Planned**: Transitions the SRDF/Star setup from concurrent SRDF to cascaded SRDF or vice versa as part of a planned event. **Unplanned**: Transitions the SRDF/Star setup from concurrent SRDF to cascaded SRDF or vice versa after a failure. Reconfigure operations can be used to resolve a transient fault or as part of a switch operation.

Required states for operations: Concurrent SRDF/Star

Normal operations

Figure 45 shows the normal operations that are available from each state.

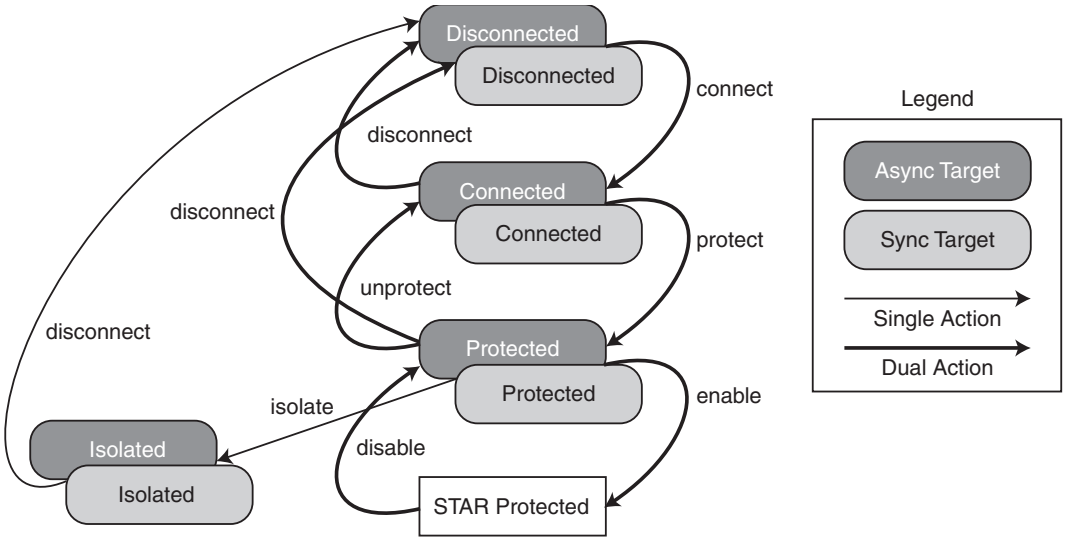


Figure 45 Concurrent SRDF/Star: Normal operations

- The `connect` operation transitions the state from Disconnected to Connected.
- The `protect` operation transitions the state from Connected to Protected.
- The `enable` operation transitions all three sites into the Star Protected state.
- The `disable`, `unprotect`, and `disconnect` operations reverse the `connect`, `protect`, and `enable` operations and revert the configuration back to the previous state.
- The `isolate` operation isolates a site and bring it down for maintenance and testing. This operation requires the Protected target site state.

Transient fault operations

Figure 46 shows the transient fault operations that are available from each state.

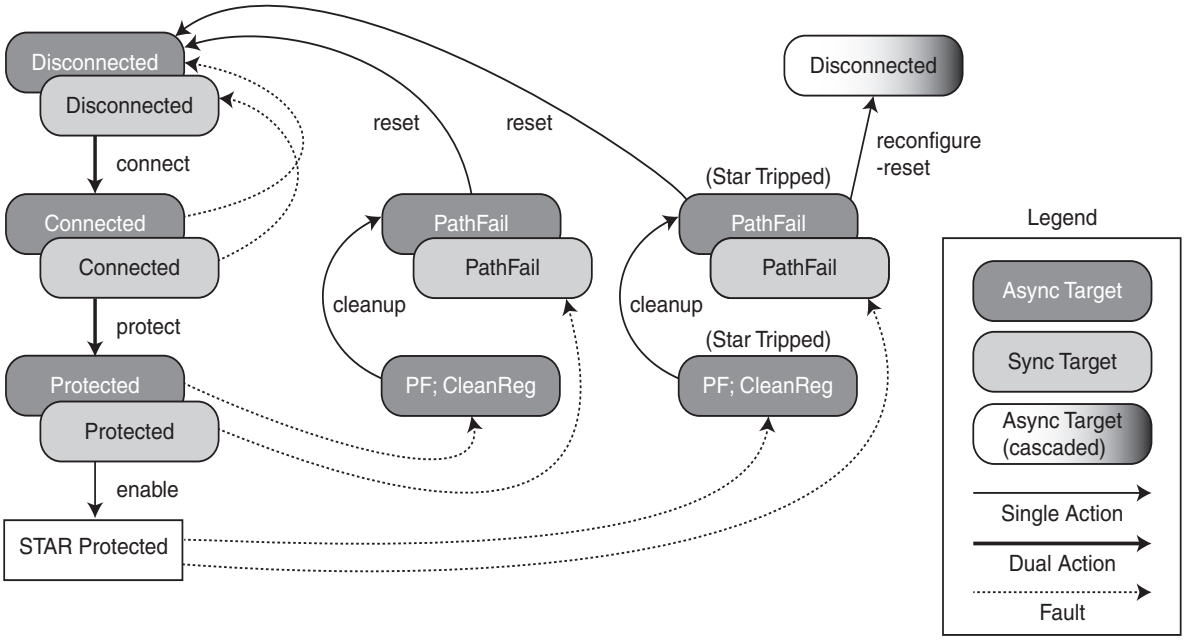


Figure 46 Concurrent SRDF/Star: Transient fault operations

After a transient fault:

- The `reset` operation transitions the state from PathFail to Disconnected.
- The `cleanup` operation performs MSC cleanup at the target site and transitions the state from PathFail;CleanReq to PathFail if the transient fault resulted from the failure of the link to the asynchronous target site.
- The `reconfigure -reset` operation changes the setup to a cascaded SRDF/Star. This operation requires that the links between the synchronous target and the asynchronous target are working.

  A reconfiguration would leave the asynchronous site in the disconnected state.
- The `connect`, `protect`, and `enable` actions bring the system to the Star Protected state.

NOTICE

EMC strongly recommends that you capture a gold copy at the failed target site after the reset action and before the `connect` operation.

Unplanned switch operations

Figure 47 shows the unplanned switch operations that are available from each state.

Note: The rounded rectangles that represent the target sites after a switch are not color coded because the definition of the workload site and the target sites can change after the switch.

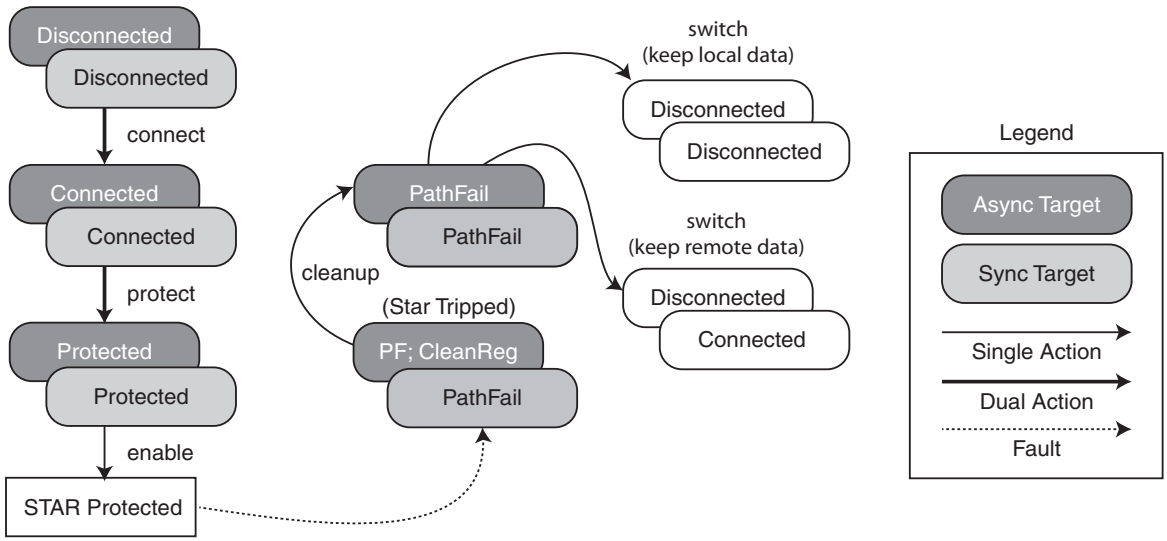


Figure 47 Concurrent SRDF/Star: Unplanned switch operations

If the workload site fails, an unplanned `switch` operation is required to move the production workload to one of the target sites.

If you switch to a target site and you:

- Keep the data at that site:
  - The `switch` operation transitions the remaining sites to the Disconnected state.
  - A `connect` operation is required to bring the sites to the Connected state.
- Keep the data at the other target site:
  - The `switch` operation transitions the other target site to the Connected state.

Planned switch operations

Figure 48 shows the planned switch operations that are available from each state.

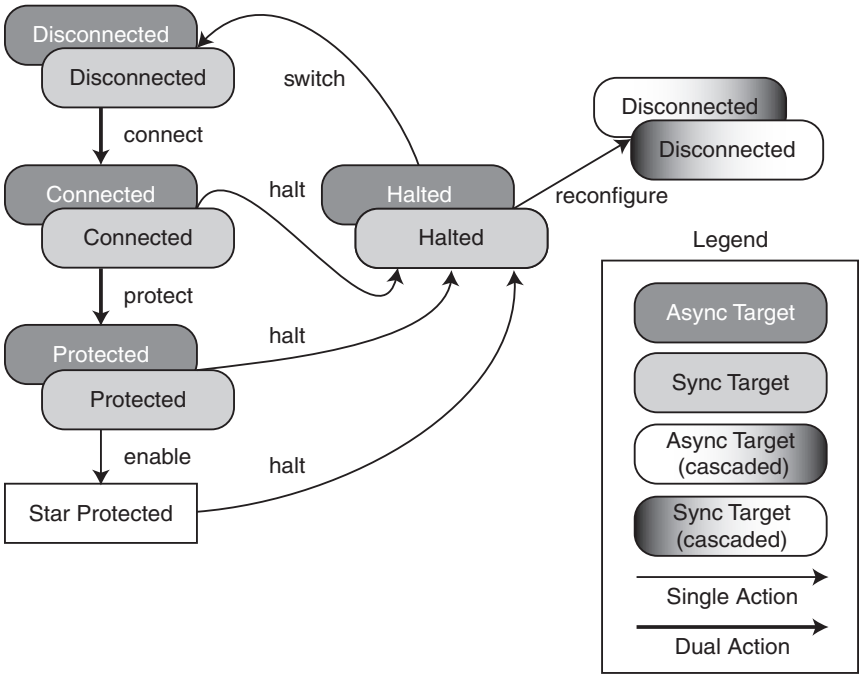


Figure 48 Concurrent SRDF/Star: Planned switch operations

IMPORTANT

Before initiating the `halt` operation, you must stop the application workload at the current workload site and unmount the file systems. If you change your mind after halting SRDF/Star, issue the `halt -reset` command to restart the workload at the current workload site.

- The `halt` operation is required for a planned switch whether you are returning the workload to the original site or moving the workload to another site.

  The `halt` operation write disables the R1 devices, drains the data to the two target sites, and makes the data at all three sites the same.

Required states for operations: Cascaded SRDF/Star

Normal operations

In cascaded SRDF/Star configurations, the consistency of the asynchronous site data is dependent on the consistency of the synchronous site data.

The asynchronous target can only be protected if the synchronous target is protected as well. After the two sites have been connected, the synchronous target must be protected first.

Note: You can isolate the synchronous target site if the asynchronous target site has a target site state of Disconnected, Isolated, or PathFail.

Figure 49 shows the normal operations that are available from each state.

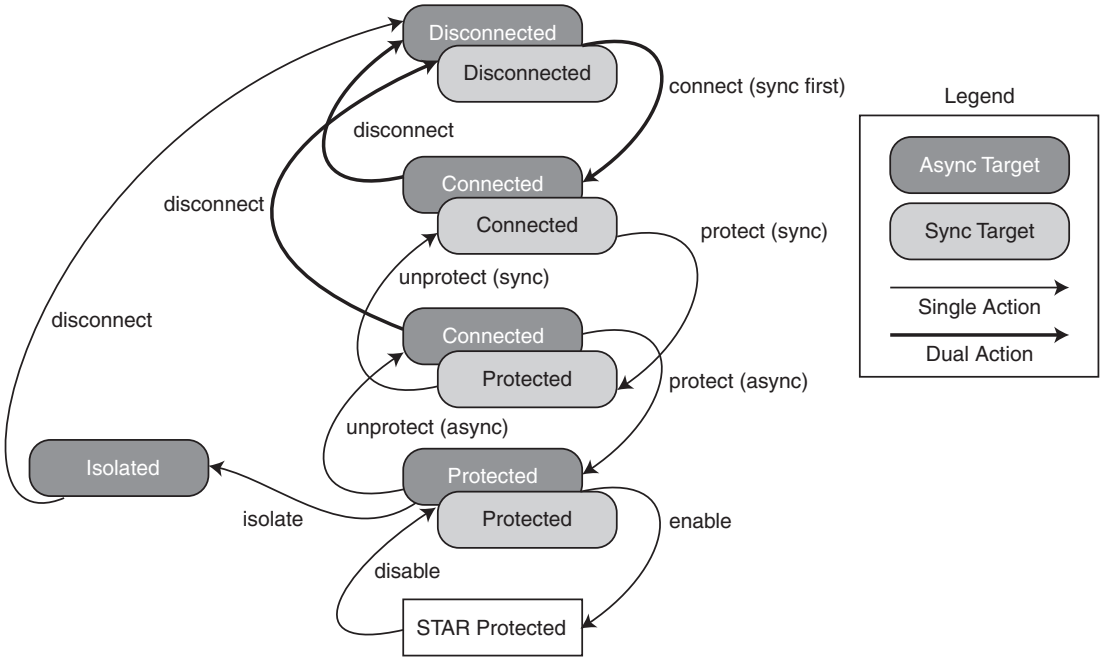


Figure 49 Cascaded SRDF/Star: Normal operations

Transient fault operations

In Cascaded SRDF/Star, the loss of either target site does not interrupt production. However, the loss of the synchronous site can result in the loss of remote replication capability (unless SRDF/Star is reconfigured to run in Concurrent SRDF/Star).

Loss of the synchronous target means that cascaded SRDF/Star is not performing replication.

If the outage is expected to be brief, you can choose to you can continue production at the workload site without remote replication. When the outage is restored, you can then reset the synchronous target.

Figure 50 shows the transient fault operations that are available from each state after the loss of the asynchronous target site.

Note: This diagram assumes that the synchronous target stayed protected during the fault.

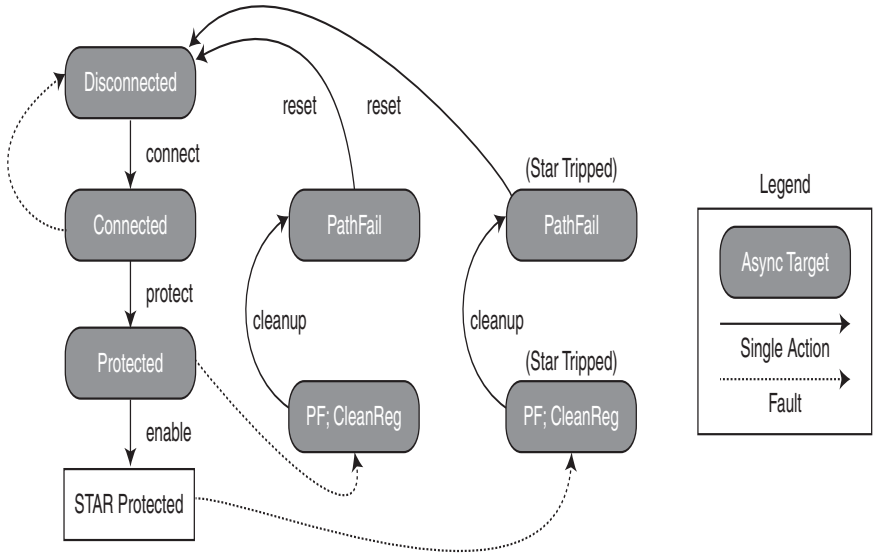


Figure 50 Cascaded SRDF/Star: Transient fault operations (asynchronous loss)

- The `reset` operation transitions the state from PathFail to Disconnected after a transient fault from the loss of the asynchronous target site.
- The `cleanup` operation (if required) performs MSC cleanup at the target site and transitions the state from PathFail;CleanReq to PathFail.

Convert Cascaded to Concurrent

Figure 51 shows the use of the `reconfigure -reset` operation to convert to Concurrent SRDF/Star with the workload site communicating directly with the asynchronous target.

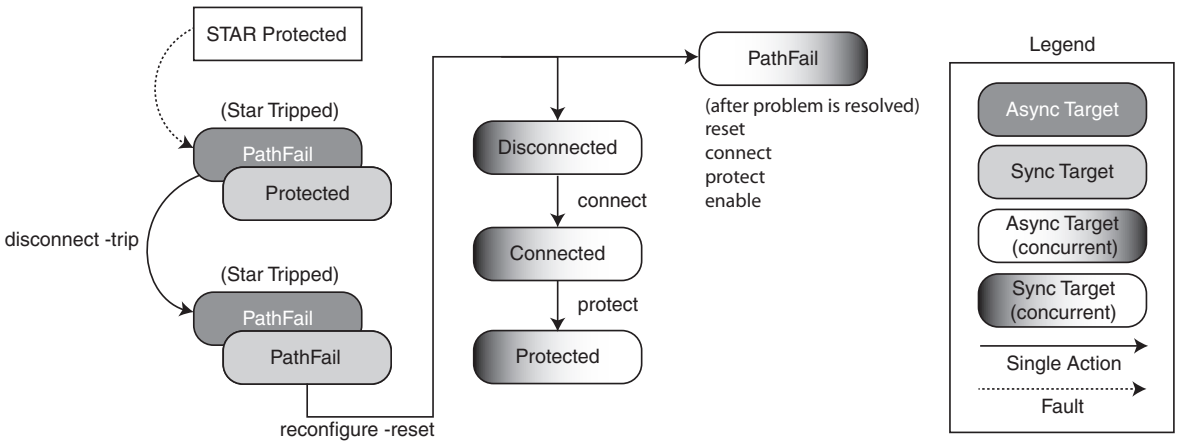


Figure 51 Cascaded SRDF/Star: Transient fault operations (synchronous loss)

If you require remote replication immediately after the synchronous target is lost, you can reconfigure Cascaded SRDF/Star to Concurrent SRDF/Star.

Unplanned switch operations

Figure 52 shows the unplanned switch operations that are available from each state.

Note: The rounded rectangles that represent the target sites after a switch are not color coded because the definition of the workload site and the target sites can change after the switch.

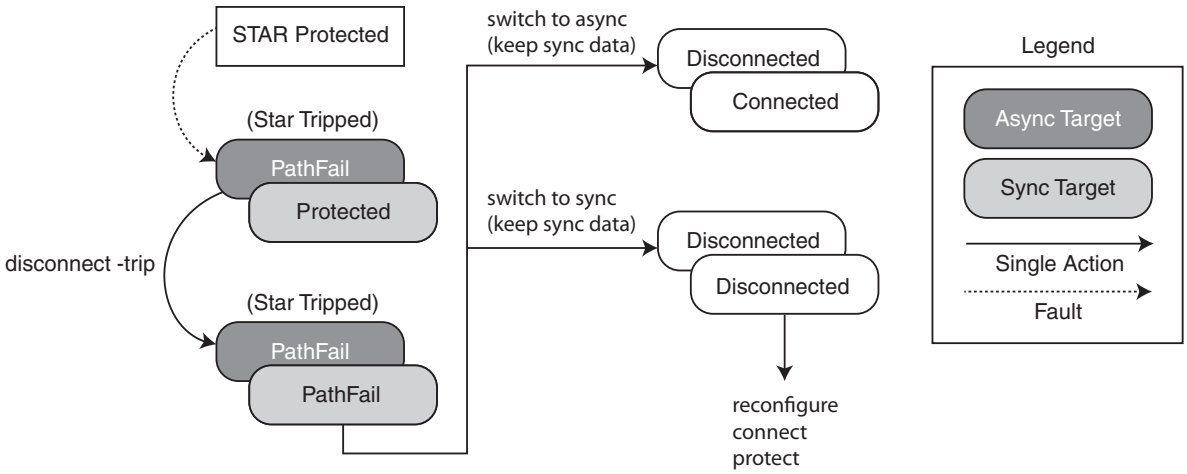


Figure 52 Cascaded SRDF/Star: Unplanned switch operations

In a Cascaded/SRDF configuration, if the workload site fails, an unplanned `switch` operation is required to move the production workload to one of the target sites.

If you switch production to the synchronous target site, you must convert the Star configuration to Concurrent SRDF/Star.

You can only keep the local data because the local data is ahead of the data at the asynchronous target site.

If you switch production to the asynchronous target site, you can keep the local data or the data at the synchronous target site.

SRDF/Star operations - summary

Table 29 SRDF/Star control operations (page 1 of 2)

Control operation	symstar argument	Description	Workload or target
Set up your SRDF/Star configuration page 236	symrdf list symcg create symcg addall symcg set symstar setup symstar buildcg symstar connect symstar protect symstar enable	Sample procedure showing the basic steps to configure and activate your SRDF/Star environment.	
Display SRDF/Star page 247 page 249 page 250	query show list	• Displays the status of a given SRDF/Star site configuration. • Displays the contents of the internal definition for a given SRDF/Star site configuration. • Lists each SRDF/Star composite group configuration, including workload name, mode of operation, CG and Star states, and target names and states.	W/T
Isolate a target site from the SRDF/Star configuration page 251	isolate	Isolates one target site from the SRDF/Star configuration and makes its R2 devices read/write enabled to their hosts.	W
Disable SRDF consistency protection for a target site page 252	unprotect	Disables SRDF/Star consistency protection to the specified target site.	W
Halt SRDF/Star page 252	halt	Used to prepare SRDF/Star for a planned switch of the workload to a target site. This action write-disables the R1 devices, drains all invalid tracks and MSC cycles so that NewYork=NewJersey=London, suspends SRDF links, disables all consistency protection, and sets adaptive copy disk mode.	W/T
Clean up after a disaster (workload site) failure page 252	cleanup	Cleans up internal meta information and cache at the remote site after a failure at the workload site.	T
Dynamically modify an SRDF/Star consistency group page 253	modifycg	Maintains consistency protection when adding or removing device pairs from an SRDF/Star consistency group.	W
Upgrade an existing SRDF/Star environment page 305	configure	Upgrades or transitions an existing SRDF/Star environment to employ R22 devices, provided the current SRDF/Star environment is operating in normal condition.	W
Begin SRDF synchronization	connect	Starts the SRDF data flow in adaptive copy disk mode.	W
Enable full SRDF/Star protection	enable	Enables complete SRDF/Star consistency protection across the three sites.	W
Enable SRDF consistency protection for a target site page 253	protect	Synchronizes devices between the workload and target sites and enables SRDF/Star consistency protection to the specified target site.	W
Change the Star replication path page 297,page 299, page 301,page 303	reconfigure	Transitions the SRDF/Star setup from concurrent SRDF to cascaded SRDF or vice versa after a site or link failure, or as part of a planned event.	W

Table 29 SRDF/Star control operations (page 2 of 2)

Control operation	symstar argument	Description	Workload or target
Reset after a transient failure page 264, page 282	reset	Cleans up internal meta information and cache at the remote site after transient fault (such as a loss of connectivity to the synchronous or asynchronous target site).	W
Switch workload operations to a target site page 268, page 289 page 293	switch	Transitions workload operations to a target site after a workload site failure or as part of a planned event.	T
Verify the given site or SRDF/Star setup is in the desired state. page 247	verify	Returns success if the state specified by the user matches the state of the Star setup.	W/T

symstar command options

Note: The `symstar` man page provides more detailed descriptions of the options used with the symstar command.

Table 30 symstar command options (page 1 of 3)

Command option	Description
`-add`	The element of configuration to add.
`-c`	Specifies the number (count) of times to display or to acquire an exclusive lock on the host database, the local array, and the remote arrays. If this option is not specified and an interval (-i) is specified, the display shows continuously, or until the SRDF/Star operation starts.
`-cg`	Name of the host composite group.
`-cg_rdfg`	The SRDF group(s) within the SRDF/Star CG in which to add or remove devices. For a concurrent SRDF/Star CG, two SRDF groups must be specified, separated by a comma. These SRDF groups are associated with the SRDF groups in the -stg_rdfg option. This association is based on their order in this option and -stg_rdfg.
`-cg_r21_rdfg`	The SRDF group connecting the R21 and R2 arrays of a cascaded SRDF/Star CG. It is only valid for operations involving cascaded R1 devices. This SRDF group is associated with the SRDF group specified in the -stg_r21_rdfg option.
`-cleanreq`	Verifies the site is in the PathFail state and needs cleaning.
`-connected`	Verifies the site is in the `connected` state.
`-devs`	Specifies the ranges of devices to add or remove.
`-disconnected`	Verifies the site is in the `disconnected` state.
`-distribute`	Performs an automatic SRDF/Star definition file distribution. This form of setup does not disrupt an active protected SRDF/Star setup.
`-full`	Used by `reconfigure`, `switch`, and `connect`. Performs a full SRDF resynchronization if SRDF incremental resync is not available. Used by the `list` action to display full names instead of abbreviations.
`-halted`	Verifies the site is in the `halted` state.
`-haltfail`	Verifies the site is in the `haltfail` state.

Table 30 symstar command options (page 2 of 3)

Command option	Description
-haltstarted	Verifies the site is in the haltstarted state.
-i	Executes a command at repeat intervals to display information or to attempt to acquire an exclusive lock on the host database, the local array, and the remote arrays. The default interval is 10 seconds. The minimum interval is 5 seconds.
-isolated	Verifies the site is in the isolated state
-keep_data	Identifies which site's data is retained when used with the switch and connect action. If you switch to the SyncTargetSite and choose to keep the data of the AsyncTargetSite, the SRDF devices are reconfigured to make a new R1?¨ R2 pairing. For the connect action, an SRDF establish or restore operation is performed, depending on which site's data is retained. By default, the workload site data is retained.
-local	Lists only the locally-defined CGs. Available only for the list action.
-offline	Obtains the data strictly from the configuration database. No connections are made to any arrays. The symstar command uses information previously gathered from the array and held in the host database as opposed to interrogating the array directly. The offline option can alternatively be set by assigning the environment variable SYMCLI_OFFLINE to 1.
-opmode	Specifies the mode of operation (concurrent or cascaded).
-path	Specifies the sites on which the new SRDF pairs are created when the reconfigure action is issued.
-pathfail	Verifies the site is in the pathfail state.
-pathfailinprog	Verifies the site is in the pathfailinprog state.
-protected	Verifies the site is in the protected state. If -site is not specified, verifies that SRDF/Star is in the protected state.
-noprompt	Suppresses the message asking you to confirm an SRDF control operation.
-reload_options	Reads the specified options file to update the SRDF/Star definition file when using the setup action. **Note:** Do not change any SITE_NAME values with this option.
-remote	Indicates the remote data copy flag. Used with the connect action when keeping remote data and the concurrent link is ready. Data is also copied to the concurrent SRDF mirror. **Note:** Not required if the concurrent link is suspended.
-remove	• For the reconfigure action, specifies the sites on which the SRDF pairs are removed. • For the setup action, specifies that all SRDF/Star mode settings for all SRDF groups be set to off if the CG is defined in the symapi database, and to remove all SRDF/Star metadata associated with the group. • For the modifycg action, indicated to remove the specified devices from the SRDF/Star CG to the staging area.
-reset	Performs a reset action on the path when the reconfigure action is issued. • When used with the halt action, allows the application to be restarted at the same site after the halt command has completed or failed. • When used with the configure action, specifies the element of the reset operation.
-site	Specifies the SiteName to apply the given action.
-stg_r21_rdfg	For modifycg operations, indicates the SRDF group comprising the staging area at the R21 array when the configuration is cascaded. Required for an add or remove operation when the setup is cascaded. This SRDF group is associated with the SRDF group in the -cg_r21_rdfg option.

Table 30 symstar command options (page 3 of 3)

Command option	Description
`-stg_rdfg`	For the `modifycg` operations, indicates the SRDF group(s) comprising the staging area. For a concurrent CG, two groups must be specified, separated by a comma. These SRDF groups are associated with the SRDF groups in the -cg_rdfg option. This association is based on their order in this option and -cg_rdfg.
`-trip`	Transitions the site to `pathfail` state when used with `disconnect` action.
`-tripped`	Verifies SRDF/Star is in the `tripped` state.
`-trip_inprogress`	Verifies SRDF/Star is in the `trip_inprogress` state.
`-unprotected`	Verifies the site is in the unprotected state. If -site is not specified, verifies SRDF/Star is in the unprotected state.
`-update`	Allows the updating of the existing host composite group from the STAR definition file.
`-v`	Provides more detailed, verbose command output.
`-wkload`	Specifies the current workload site name if symstar fails to determine the current workload site name.

Command failure while in Connected state

While in the SRDF/Star Connected state, if an operation fails that indicates the SRDF mode is invalid, issue the `symstar configure -reset rdf_mode` command at the workload site.

This command resets the device pairs in the SRDF/Star CG to adaptive copy, and if the composite group has R22 devices, the SRDF mode for the recovery pairs is also set to adaptive copy.

Restrictions for cascaded mode

- `symstar protect` command to the asynchronous target is allowed only if the synchronous target site is in a *Protected* state.

  An unprotected flow of data is not allowed from the workload site to the synchronous target site if the asynchronous target site is in a *Protected* state as this will result in an inconsistent data image at the asynchronous target site.

- If the asynchronous target site is in a *Protected* state, the `symstar connect` and `symstar unprotect` commands are not allowed to the synchronous target site as this will also result in an inconsistent data image at the asynchronous target site.

- `symstar isolate` command cannot be issued to the synchronous target site since it is not possible to isolate only the synchronous target site

The remainder of this chapter consists of detailed descriptions and command examples for the operations listed in Table 29.

Configure and bring up SRDF/Star

This section lists the steps to configure and bring up the SRDF/Star environment and links to detailed instructions for each step:

1. Verify the SRDF/Star host is locally connected to *only* one of the three sites.

   "Step 1: Verify SRDF/Star host connectivity."

2. Verify the settings for each array to be included in the SRDF/Star configuration.

   "Step 2: Verify array settings."

3. Create a composite group at the workload site.

   "Step 3: Create an SRDF/Star composite group" on page 237.

4. Create an SRDF/Star options file containing specific parameters for the setup procedure.

   "Step 4: Create the SRDF/Star options file" on page 242.

5. Use the SRDF/Star `symstar setup` command to read and validate the information in the host composite group definition, and build the SRDF/Star definition file that defines the R1 composite group.

   "Step 5: Perform the symstar setup operation" on page 243.

6. Optionally, use the `symstar buildcg` command to build matching R2 or R21 composite groups at the target sites.

   "Step 6: Create R2 or R21 composite groups" on page 245.

7. Optionally, add BCVs to the SRDF/Star configuration.

   "Step 7: (optional) Add BCV devices to the SRDF/Star configuration" on page 245.

8. Bring up your SRDF/Star configuration.

   "Step 8: Bring up the SRDF/Star configuration" on page 246

To perform SRDF/Star operations with access control enabled, you need the SRDF, BASECTRL, BASE, and BCV access types.

EMC Solutions Enabler Array Management CLI User Guide provides more information.

Note: An SRDF/Star environment contains one or more triangles, where each triangle has a unique SRDF group for the synchronous link, the asynchronous link, and the recovery group link. No sharing of SRDF groups is allowed between any two SRDF/Star triangles.

The examples in this section use the following names:

- `StarGrp` - the composite group and
- `NewYork` - workload site
- `NewJersey` - synchronous target site
- `London` - asynchronous target site

Step 1: Verify SRDF/Star host connectivity

The SRDF/Star host must be connected locally to *only* one of the three sites.

Use the `symcfg list` command to verify the configuration.

The following output displays the required connectivity of `Local`, `Remote`, `Remote` under `Attachment`:

```
symcfg list

                         S Y M M E T R I X

                                  Mcode    Cache      Num Phys  Num Symm
SymmID       Attachment  Model    Version  Size (MB)  Devices   Devices

000194901217 Local       VMAX-1SE 5876       28672       369      6689
000194901235 Remote      VMAX-1SE 5876       28672         0      6890
000194901241 Remote      VMAX-1SE 5876       28672         0      7007
```

Step 2: Verify array settings

- Verify that each array within SRDF/Star uses dynamic SRDF devices.

  Use the `symrdf list` command with the -dynamic option to display SRDF devices configured as dynamic SRDF-capable.

- Verify that the SRDF directors are Fibre or GigE (RF or RE).

  Use the `symcfg list -sid SID -rdfg all` command to display SRDF group-level settings for a specific group or all groups including director configuration.

- Use the `symcfg list -v` command to verify that the following states exist for each array within SRDF/Star:

  - Concurrent SRDF Configuration State = Enabled
  - Dynamic SRDF Configuration State = Enabled
  - Concurrent Dynamic SRDF Configuration = Enabled
  - RDF Data Mobility Configuration State = Disabled

- Use the `symcfg list -rdfg -v` command to verify that each SRDF group in the composite group has the following configuration:

  - Prevent RAs Online Upon Power On = Enabled
  - Prevent Auto Link Recovery = Enabled

  Note: Preventing automatic recovery preserves the remote copy that was consistent at the time of the link failure.

Step 3: Create an SRDF/Star composite group

This step includes the following tasks:

- Create an RDF1 type composite group on the control host for the array at the workload site (NewYork).
- Enable consistency protection for the composite group.
- Populate the consistency group with devices.

This step varies depending on the topology of your SRDF configuration:

- If your topology is concurrent, proceed to "Create a composite group in concurrent SRDF/Star" on page 238.
- If your topology is cascaded, skip to "Create a composite group in cascaded SRDF/Star" on page 240.

Create a composite group in concurrent SRDF/Star

Follow these steps if your SRDF/Star configuration is a concurrent topology.

The following example procedure includes:

- A composite group named `StarGrp`
- The workload site is `NewYork`
- The synchronous target site is `NewJersey`, and
- The asynchronous target site is `London`.

Figure 53 shows the concurrent topology:

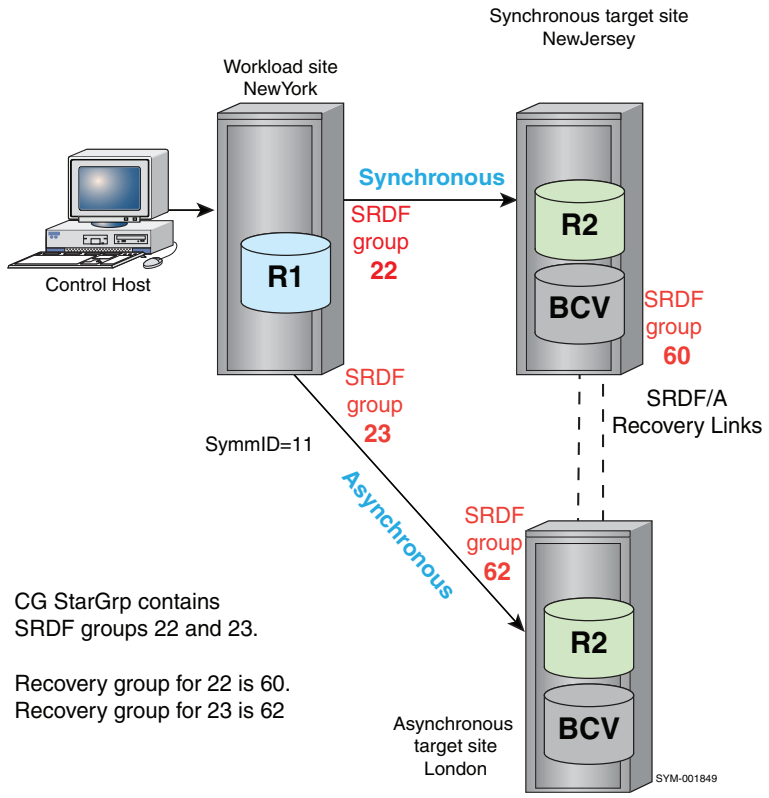


Figure 53 Concurrent SRDF/Star setup using the StarGrp composite group

---

Note: *EMC Solutions Enabler Array Management CLI User Guide* provides additional information on composite groups and using the `symcg -cg` command.

Use the following steps to build an RDF1 type composite group on the control host of the SRDF/Star workload site (`NewYork`, `SID 11`) in a concurrent configuration:

1. Determine which devices on the local array are configured as concurrent dynamic devices.

To list the concurrent dynamic devices for array 11:

```
symrdf list -sid 11 -concurrent -dynamic -both
```

Note: Use the `-dynamic` and `-both` options to display dynamic SRDF pairs in which the paired devices can be either R1 or R2 devices.

2. Create an RDF1-type composite group with consistency protection on the control host at the workload site.

   To create composite group `StarGrp` on array `NewYork`:

   ```
   symcg create StarGrp -type rdf1 -rdf_consistency
   ```

   Note: The `-rdf_consistency` option specifies consistency protection for the group.

3. Add devices to the composite group from those SRDF groups that represent the concurrent links for the SRDF/Star configuration.

   To add all the devices in SRDF groups 23 and 22 to composite group `StarGrp`:

   ```
   symcg -cg StarGrp -sid 11 addall dev -rdfg 23
   ```

   Note: With concurrent SRDF, the command that adds one of the two concurrent groups adds both concurrent groups (in this example, the synchronous SRDF group 22 is automatically added with the asynchronous SRDF group 23).

4. Create two SRDF group names; one for all synchronous links and one for all asynchronous links.

   To create two SRDF group names NewJersey for SRDF Group 22 on SID 11and SRDF group name London for SRDF group 23 on SID 11:

   ```
   symcg -cg StarGrp set -name NewJersey -rdfg 11:22
   symcg -cg StarGrp set -name London -rdfg 11:23
   ```

   Note: You could include additional synchronous SRDF groups in (synchronous) NewJersey using the `sid:rdfg` syntax. The site named `London` includes concurrent asynchronous SRDF group 23.

   You must also include the names `NewJersey` and `London` in the SRDF/Star options file as the values for the synchronous and asynchronous target site names, respectively.

   "Step 4: Create the SRDF/Star options file" on page 242 provides more information.

5. For each source SRDF group that you added to the composite group, define a corresponding recovery SRDF group at the remote site.

   A recovery SRDF group can be static or dynamic, but it cannot be shared. A recovery SRDF group cannot contain any devices.

   In the following example:

- SRDF group 60 is an empty static or dynamic group on the remote array to which source SRDF group 22 is linked.
- Recovery SRDF group 62 was configured on the other remote array as a match for the source SRDF group 23.

To set the remote recovery group for StarGp RDF group 22 to SRDF group 60 at the remote site:

```
symcg -cg StarGrp set -rdfg 11:22 -recovery_rdfg 60
```

To set the remote recovery group for StarGp RDF group 23 to SRDF group 62 at the remote site:

```
symcg -cg StarGrp set -rdfg 11:23 -recovery_rdfg 62
```

Note: These two recovery group definitions represent one recovery SRDF group as viewed from each of the two target sites.

6. Skip to "Step 4: Create the SRDF/Star options file" on page 242.

Create a composite group in cascaded SRDF/Star

Follow these steps if your SRDF/Star configuration is a cascaded topology.

The following example procedure includes:

- A composite group named `StarGrp`
- The workload site is `NewYork`
- The synchronous target site is `NewJersey`, and
- The asynchronous target site is `London`.

Figure 53 shows the cascaded topology:

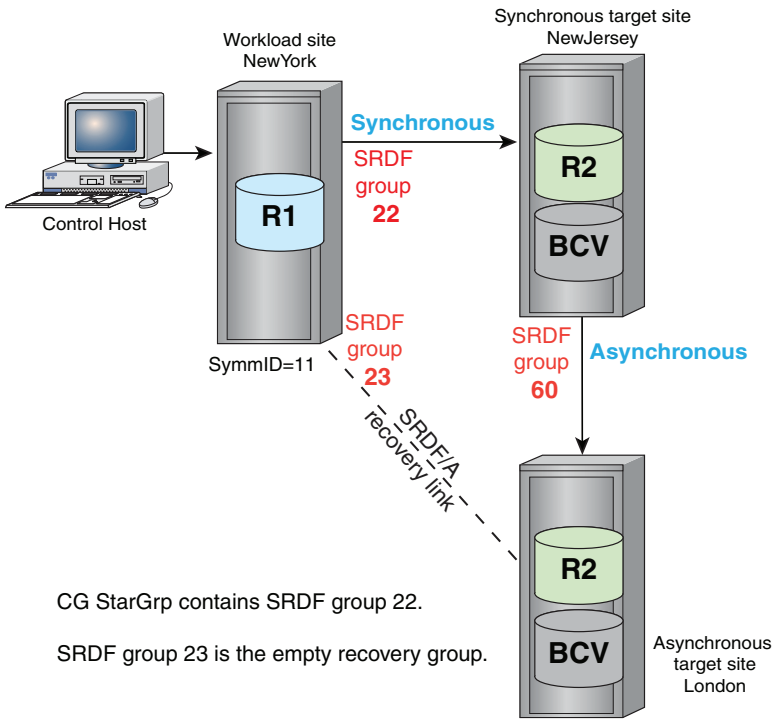


Figure 54 Cascaded SRDF/Star setup using the StarGrp composite group

Use the following steps to build an RDF1type composite group on the control host of the SRDF/Star workload site (`NewYork`, `SID 11`) in a cascaded environment:

1. Determine which devices on the local array (`-sid 11`) are configured as cascaded dynamic devices.

   To list the cascaded dynamic devices for array 11:

   ```
   symrdf list -sid 11 -R1 -cascaded -dynamic -both
   ```

   Note: Use the `-dynamic` and `-both` options to display dynamic SRDF pairs in which the paired devices can be either R1 or R2 devices.

2. Create an RDF1-type composite group with consistency enabled on the control host at the workload site.

   To create composite group `StarGrp` with consistency enabled at `NewYork`:

   ```
   symcg create StarGrp -type rdf1 -rdf_consistency
   ```

   Note: Use the `-rdf_consistency` option to specify consistency protection for the group.

3. Add devices to the composite group from those SRDF groups that represent the cascaded links for the SRDF/Star configuration.

   To add devices in SRDF group 22 to composite group StarGrp:

   ```
   symcg -cg StarGrp -sid 11 addall dev -rdfg 22
   ```

4. Create one SRDF group name for all synchronous links.

   To create SRDF group name NewJersey for devices in SRDF group 22 on SID 11:

   ```
   symcg -cg StarGrp set -name NewJersey -rdfg 11:22
   ```

   Note: The site named `NewJersey` includes synchronous SRDF group 22 on array 11. Use the `sid:rdfg` syntax to add additional synchronous groups.

   You must include the site names `NewJersey` and `London` in the SRDF/Star options file as the values for the synchronous and asynchronous target site names, respectively. "Step 4: Create the SRDF/Star options file" on page 242 provides more information.

5. For each source SRDF group that you added to the composite group, define a corresponding recovery SRDF group at the local (workload) site.

   The recovery SRDF group:

   - Can be static or dynamic
   - Cannot be shared.
   - Cannot contain any devices.
   - Must be empty.

For the cascaded setup in Figure 54 on page 241, the recovery SRDF group is the empty SRDF group 23 configured between the NewYork sync site and the London asynchronous site.

To add this recovery SRDF group:

```
symcg -cg StarGrp set -rdfg 11:22 -recovery_rdfg 23
```

Step 4: Create the SRDF/Star options file

The SRDF/Star options file specifies the names of each SRDF/Star site and other required parameters.

You can add comment lines that begin with"#", as shown below. This file must conform to the following syntax:

```
SYMCL_STAR_OPTION=Value
```

Example

```
#Comment
SYMCLI_STAR_WORKLOAD_SITE_NAME=WorkloadSiteName
SYMCLI_STAR_SYNCTARGET_SITE_NAME=SyncSiteName
SYMCLI_STAR_ASYNCTARGET_SITE_NAME=AsyncSiteName
SYMCLI_STAR_ADAPTIVE_COPY_TRACKS=NumberTracks
SYMCLI_STAR_ACTION_TIMEOUT=NumberSeconds
SYMCLI_STAR_TERM_SDDF=Yes|No
SYMCLI_STAR_ALLOW_CASCADED_CONFIGURATION=Yes|No
SYMCLI_STAR_SYNCTARGET_RDF_MODE=ACP|SYNC
SYMCLI_STAR_ASYNCTARGET_RDF_MODE=ACP|ASYNC
```

Note: If the options file contains the `SYMCLI_STAR_COMPATIBILITY_MODE` parameter, it must be set to `v70`.

- **`WorkloadSiteName`** - Configure a meaningful name for the workload site.
- **`SyncSiteName`** - Configure a meaningful name for the synchronous target site.

  This name must match the SRDF group name used for the synchronous SRDF groups when building the composite group.
- **`AsyncSiteName`** - Configure a meaningful name for the asynchronous target site.

  This name must match the SRDF group name that you used for the asynchronous SRDF groups when building the composite group for a concurrent STAR configuration.

  There are no SRDF group names for the asynchronous site in a cascaded configuration.
- **`NumberTracks`** - Maximum number of invalid tracks allowed for Star to transition from adaptive copy mode to synchronous or asynchronous mode. Star will wait until the number of invalid tracks is at or below `NumberTracks` before changing the SRDF mode.

  The default is 30,000.
- **`NumberSeconds`** - Maximum time (in seconds) that the system waits for a particular condition before returning a time-out failure.

  The wait condition may be the time to achieve R2-recoverable SRDF/Star protection or SRDF consistency protection, or the time for SRDF devices to reach the specified number of invalid tracks while synchronizing.

  The default is 1800 seconds (30 minutes).

The smallest value allowed is 300 seconds (5 minutes).

- **SYMCLI_STAR_TERM_SDDF** – Enables/disables termination of Symmetrix Differential Data Facility sessions on both the SYNC and ASYNC target sites during a `symstar disable`.

  Yes - Terminates SDDF sessions during a `symstar disable`.

  No - (default setting) Deactivates (instead of terminates) the SDDF sessions during a `symstar disable`.

- **SYMCLI_STAR_ALLOW_CASCADED_CONFIGURATION** – Enables/disables STAR mode for cascaded SRDF/Star configurations.

  `Yes` - STAR mode for a cascaded SRDF/Star configuration.

  `No` - (default setting).

- **SYMCLI_STAR_SYNCTARGET_RDF_MODE** - Sets the SRDF mode between the workload site and the synchronous target site at the end of the `symstar unprotect` operation.

  `ACP` - (default setting) Sets the SRDF mode between the workload site and the synchronous target site transitions to adaptive copy mode at the end of the `symstar unprotect` operation.

  `SYNC` - Sets the SRDF mode between the workload site and synchronous target site remains synchronous at the end of the `symstar unprotect` action.

- **SYMCLI_STAR_ASYNCTARGET_RDF_MODE** - Sets the SRDF mode between the workload site and the asynchronous target site at the end of the `symstar unprotect` operation.

  `ACP` - (default setting) Sets the SRDF mode between the workload site and the asynchronous target site to transition to adaptive copy mode at the end of the `symstar unprotect` operation.

  `ASYNC` - The SRDF mode between the workload site and asynchronous target site remains asynchronous at the end of the `symstar unprotect` action.

Sample options file

The following sample options file defines sites in `NewYork`, `NewJersey`, and `London` as operating points of a company's concurrent SRDF/Star storage environment:

```
#ABC Company's April 2012 financial Star storage environment
SYMCLI_STAR_WORKLOAD_SITE_NAME=NewYork
SYMCLI_STAR_SYNCTARGET_SITE_NAME=NewJersey
SYMCLI_STAR_ASYNCTARGET_SITE_NAME=London
SYMCLI_STAR_ADAPTIVE_COPY_TRACKS=30000
SYMCLI_STAR_ACTION_TIMEOUT=1800
SYMCLI_STAR_TERM_SDDF=No
SYMCLI_STAR_ALLOW_CASCADED_CONFIGURATION=No
SYMCLI_STAR_SYNCTARGET_RDF_MODE=ACP
```

Step 5: Perform the symstar setup operation

The SRDF/Star `symstar setup` command:

- Reads and validates the information in the host composite group definition, and
- Builds the SRDF/Star definition file that defines the R1 consistency group for the workload site.

This information is combined with the settings in your SRDF/Star options file, and then automatically written in an internal format to the SFS on a array at each site.

symstar setup command syntax

The following is the syntax for the `symstar setup` command:

```
symstar -cg CgName
   setup -options OptionsFile [-distribute]
      [-site SiteName]
      [-opmode <concurrent | cascaded>]

   setup -options OptionsFile -reload_options

   setup -remove [-force]
```

Where:

- `-reload_options` - updates the options values in the SRDF/Star definition file.

  Note: Do not use this option to update any site name values.

- `setup -remove` - changes the STAR mode setting of all participating SRDF groups to OFF and removes the SRDF/Star definition files from all reachable sites.
- `setup -options OptionsFile` - validates the specified host composite group definition and builds the file that defines the R1 consistency group for the workload site.
- `-distribute` - distributes SRDF/Star CGs in which the SRDF/Star definition file was manually copied to each site.

  This option automatically distributes the SRDF/Star definition file to a array at each site without altering the state of the SRDF/Star setup.

  Note: Use the `-distribute` option from the workload site when both target sites are reachable.

Usage notes

- Use the `setup -remove` option from the workload site and when the target sites are either in the Connected or Disconnected state.

  Note: SRDF/Star must be disabled with both target sites in the Unprotected state.

- You can run `setup -remove -force` from a non-workload site when the remote sites are in the PathFail state or in a STAR Tripped state.
- The `setup -remove -force` command removes all distributed SRDF/Star definition files associated with an SRDF/Star consistency group even when its definition no longer exists in the SYMAPI database. It also removes the host's local definition files for the SRDF/Star CG.
- If a site is unreachable, you must run the `setup -remove -force` command at that site to remove the SRDF/Star definition file from the SFS, and remove the host's local definition files of the SRDF/Star CG.

Example To build the definition file for the `StarGrp` CG using the settings from the options file created in Step 4 (`MyOpFile.txt`):

```
symstar -cg StarGrp setup -options MyOpFile.txt
```

Step 6: Create R2 or R21 composite groups

Once the setup is complete and the SRDF/Star definition file is distributed to the SFS at the other sites, use the `symstar buildcg` command to create the matching R2 or R21 composite groups needed for recovery operations at the synchronous and asynchronous target sites.

symstar buildcg command syntax

```
symstar -cg CgName [-noprompt]
   buildcg -site SiteName [-update]
```

To create the matching composite groups for NewJersey and London:

```
symstar -cg StarGrp buildcg -site NewJersey
symstar -cg StarGrp buildcg -site London
```

Usage notes

- The `setup` and `buildcg` actions ignore BCV devices that you may have added to the composite group at the workload site (`NewYork`).
- If remote BCVs are protecting data during the resynchronization of the synchronous and asynchronous target sites, you must manually add the BCVs to the synchronous and asynchronous composite groups.

The next step varies depending on whether you are using BCV devices:

- If you are using BCV devices to retain a consistent restartable image of the data, proceed to "Step 7: (optional) Add BCV devices to the SRDF/Star configuration."
- If you are not using BCV devices, skip to "Step 8: Bring up the SRDF/Star configuration."

Step 7: (optional) Add BCV devices to the SRDF/Star configuration

BCVs retain a consistent restartable image of the data volumes during periods of resynchronization.

BCVs are optional, but strongly recommended at both the synchronous and asynchronous target sites (`NewJersey` and `London`).

Use the following steps to add BCV devices to the SRDF/Star configuration:

1. Add BCVs at the remote target sites by associating the BCVs with the composite group.

   To associate the BCVs with the composite group StarGrp:

```
symbcv -cg StarGrp -sid 11 associateall dev -devs 182:19A -rdf -rdfg 22
```

   To associate the BCVs with the composite group StarGrp in a concurrent SRDF/Star configuration:

```
symbcv -cg StarGrp -sid 11 associateall dev -devs 3B6:3C9 -rdf -rdfg 23
```

Note: Include the SRDF group number of the local R1 source devices.

2. Use the following commands to synchronize the remote BCV pairs. Data is copied from the R2 or R21 devices on the remote arrays to the BCV devices there.

   The `-rdf` option identifies the targets as the remote BCVs.

   The names `NewJersey` and `London` are those that were previously set for SRDF groups 22 and 23 (concurrent SRDF/Star setup only), respectively.

   The `-star` option is required for any TimeFinder operations that affect BCV devices in an SRDF/Star composite group. To synchronize the remote BCV pairs.

   ```
   symmir -cg StarGrp establish -star -full -rdf -rdfg name:NewJersey
   symmir -cg StarGrp establish -star -full -rdf -rdfg name:London
   ```

Note: You can associate BCVs to a composite group either before or after performing the `setup` operation. The `setup` operation does not save BCV information for the composite group, so any BCVs that were associated are excluded from the internal definitions file copied to the remote hosts.

Step 8: Bring up the SRDF/Star configuration

1. Use the `symstar query` command to determine if the setup action left the target sites in a *Connected* or *Disconnected* state.

   To query SRDF group StarGrp:

   ```
   symstar -cg StarGrp query -detail
   ```

   Note: "symstar show command" on page 249 provides an example of the output returned with this command.

2. The next step varies depending on whether the system state is *Connected* or *Disconnected.* For both concurrent and cascaded environments:

   If the system state is:

   - *Connected* - The devices are already read/write (RW) on the SRDF link.

     Skip to step 3 .

   - *Disconnected* - Use the following commands to connect SRDF/Star: first `NewJersey` and then `London`.:

     ```
     symstar -cg StarGrp connect -site NewJersey
     symstar -cg StarGrp connect -site London
     ```

3. Use the following commands to bring up SRDF/Star: first `NewJersey` and then `London`:

   ```
   symstar -cg StarGrp protect -site NewJersey
   symstar -cg StarGrp protect -site London
   symstar -cg StarGrp enable
   ```

Where:

- The `connect` action sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links, but does not wait for synchronization.
- The `protect` action transitions to the correct SRDF mode (sync or async), enables SRDF consistency protection, waits for synchronization, and sets the STAR mode indicators.
- The `enable` action provides complete SRDF/Star protection, including:
  - Creates and initializes the SDDF sessions,
  - Sets the STAR mode indicators on the recovery groups,
  - Enables SRDF/Star to wait for R2-recoverable STAR protection across SRDF/S and SRDF/A before producing a STAR Protected state.

Note: To bring up London and then NewJersey in a concurrent SRDF/Star configuration, reverse the order of the `symstar protect` commands.

Displaying your symstar configuration

This section describes output of the following:

- "symstar query command"
- "symstar show command"
- "symstar list command"

See also:

- "Commands to display, query and verify SRDF configurations" on page 34
- "symrdf list command options" on page 38

symstar query command

Use the `symstar query` command to display the local and remote array information and the status of the SRDF pairs in the composite group.

To display the status of the SRDF/Star site configuration for a composite group called `StarGrp`, enter:

```
symstar query -cg StarGrp

Site Name                                       : NewYork

Workload Site                                   : NewYork
1st Target Site                                 : NewJersey
2nd Target Site                                 : London

Composite Group Name                            : StarGrp
Composite Group Type                            : RDF1
Composite Group State                           : Valid

Workload Data Image Consistent                  : Yes
System State:
  {
  1st_Target_Site                               : Protected
  2nd_Target_Site                               : Protected
```

```
  STAR                                          : Protected
  Mode of Operation                             : Concurrent
  }

Last Action Performed                           : Enable
Last Action Status                              : Successful
Last Action Timestamp                           : 10/15/2010_16:07:39

STAR Information:
  {
  STAR Consistency Capable                      : Yes
  STAR Consistency Mode                         : STAR
  Synchronous Target Site                       : NewJersey
  Asynchronous Target Site                      : London
  Differential Resync Available                 : Yes
  R2 Recoverable                                : Yes
  Asynchronous Target Site Data most Current    : No
  }

1st Target Site Information:
  {
  Source Site Name                              : NewYork
  Target Site Name                              : NewJersey
  RDF Consistency Capability                    : SYNC
  RDF Consistency Mode                          : SYNC
  Site Data Image Consistent                    : Yes

            Source Site                      Target Site
  ------------------------------ -- ------------------------------ - ------------
            ST                   LI           ST                   M
        RD   A                    N Rem   RD   A                   O
  Symm   F   T   R1 Inv   R2 Inv  K Symm   F   T   R1 Inv   R2 Inv D RDF Pair
  ID     G   E   Tracks   Tracks  S ID     G   E   Tracks   Tracks E STATE
  ----- --- -- -------- -------- -- ----- --- -- -------- -------- - ------------

  02011  22 RW        0        0 RW 00016 150 WD        0        0 S Synchronized

   Totals:  -- -------- -------- --           -- -------- -------- - ------------
            RW        0        0 RW           WD        0        0 S Synchronized
  }

2nd Target Site Information:
  {
  Source Site Name                              : NewYork
  Target Site Name                              : London
  RDF Consistency Capability                    : MSC
  RDF Consistency Mode                          : MSC
  Site Data Image Consistent                    : Yes

            Source Site                      Target Site
  ------------------------------ -- ------------------------------ - ------------
            ST                   LI           ST                   M
        RD   A                    N Rem   RD   A                   O
  Symm   F   T   R1 Inv   R2 Inv  K Symm   F   T   R1 Inv   R2 Inv D RDF Pair
  ID     G   E   Tracks   Tracks  S ID     G   E   Tracks   Tracks E STATE
  ----- --- -- -------- -------- -- ----- --- -- -------- -------- - ------------

  02011  23 RW        0        0 RW 00109 145 NR        0        0 A Consistent

   Totals:  -- -------- -------- --           -- -------- -------- - ------------
            RW        0        0 RW           NR        0        0 A Consistent
  }

Legend:

 Modes:
```

```
Mode of Operation: A=Async, C=Adaptive Copy, S=Sync, O=Other, M=Mixed
```

Note: Using the `-detail` option with `query` includes extended information, such as the full Symmetrix IDs, status flags, recovery SRDF groups, and SRDF mode in the output.

symstar show command

Use the `symstar show` command to display the contents of the SRDF/Star definition file that was created by the `symstar setup` command. To display the SRDF/Star definition file for the `StarGrp` composite group, enter:

```
symstar -cg StarGrp show

Composite Group Name          : StarGrp

Recovery RDF Pairs configured : Yes
Diskless Device Site          : N/A

 Site NewYork to site NewJersey Information:

 ----------------------------------
  Workload   View  SyncTarget  View
 ----------------  ----------------
              RD                RD
 Symmetrix     F  Symmetrix     F
 ID            G  ID            G
 ------------ --  ------------ ---
 000190102011 22 000190300016    8
 --------------------------------------

 Site NewYork to site London Information:
 {
 --------------------------------------
  Workload   View  ASyncTarget  View
 ----------------  ----------------
              RD                RD
 Symmetrix     F  Symmetrix     F
 ID            G  ID            G
 ------------ --  ------------ ---
 000190102011 23 000190300109   14
 ---------------------------------------

 Site NewJersey to site London Information:

 --------------------------------------
  SyncTarget  View  ASyncTarget  View
 ----------------- ----------------
              RD                 RD
 Symmetrix     F   Symmetrix     F
 ID            G   ID            G
 ------------ --  ------------ ---
 000190300016 60 000190300109   62
 ---------------------------------------

Options file settings                 :

      WorkloadSite                        : NewYork
      SyncTargetSite                      : NewJersey
      AsyncTargetSite                     : London
      Adaptive_Copy_Tracks                : 30000
      Action_Timeout                      : 1800
      Term_Sddf                           : Yes
      Allow_Cascaded_Configuration        : No
      Star_Compatibility_Mode             : v70
```

```
Auto_Distribute_Internal_File        : Yes
SyncTarget_RDF_Mode                  : ACP
AsyncTarget_RDF_Mode                 : ASYNC
```

Note: To display all the devices with SRDF/Star, include the `-detail` option.

symstar list command

The `symstar list` command displays configuration information about the SRDF/Star composite groups that have the SRDF/Star definition file defined locally or on locally-attached SFS devices.

To list the configurations for all the SRDF/Star composite groups, enter:

symstar list

```
S T A R   G R O U P S
-------------------------------------------------------------------------------
                                          First Target      Second Target
                  Flags Workload   Star   ----------------- -----------------
Name               MLC  Name       State  Name       State  Name       State
-------------------------------------------------------------------------------
abc_test_cg_1      CW.  MyStarSit* Unprot MyStarSit* Conn   MyStarSit* Disc
boston_grp         CFV  Hopkinton  Trip   Westborou* Pfl    Southboro* Pfl
citi_west          CFV  Site_A     Unprot Site_B     Disc   Site_C     Conn
ha_apps_cg         CS.  Boston     Unprot Cambridge  Haltst SouthShor* Haltfl
ny                 CW.  A          Unprot B          Halt   C          Halt
star_cg            AS.  Boston     Prot   NewYork    Prot   Philly     Prot
ubs_core           AFI  A_Site     Trip   B_Site     Pfl    C_Site     Pfl
zcg                AW.  SITEA      -      SITEB      -      SITEC      -
zcg2               ..I  -          -      -          -      -          -
zcg3               ..I  -          -      -          -      -          -

Legend:

  Flags:
    M(ode of Operation) : C = Concurrent, A = Cascaded, . = Unknown
    L(ocal Site)        : W = Workload, F = First target,
                          S = Second target, . = Unknown
    C(G State)          : V = Valid, I = Invalid, R = RecoveryRequired,. = Not defined

  States:
    Star State          : Prot = Protected, Prprot = PartiallyProtected,
                          Trip = Tripped, Tripip = TripInprogress,
                          Unprot = Unprotected, - = Unknown

    Target State        : Conn = Connected, Disc = Disconnected, Halt = Halted,
                          Haltfl = HaltFail, Haltst = HaltStarted,
                          Isol = Isolated, Pfl = PathFail, Prot = Protected,
                          Pflip = PathFailInProg, Pflcl = Pathfail CleanReq,
                          - = Unknown
```

An entry containing a dash or a dot in the `symstar list` output indicates the command was unable to determine this value.

Basic SRDF/Star operations

This section describes the following topics:

- "Isolating the SRDF/Star sites"
- "Unprotecting the target sites"
- "Halting the target sites"
- "Cleanup metadata"

Isolating the SRDF/Star sites

There may be occasions when you want to isolate one of the SRDF/Star sites, perhaps for testing purposes, and then rejoin the isolated site with the SRDF/Star configuration. The `symstar isolate` command allows you to temporarily isolate one or all of the SRDF/Star sites.

Note: In rejoining an isolated site to the SRDF/Star configuration, any updates made to London's R2 devices while *isolated* are discarded. That is, the data on the R1 devices overwrites the data on the R2 devices.

The `symstar isolate` command has the following requirements:

- SRDF/Star protection must be disabled.
- The site to be isolated must be in the Protected state.
- If there are BCVs at the target site that are paired with the SRDF/Star R2 devices, split these BCV pairs before executing the command.

Note: In a cascaded SRDF/Star configuration you cannot isolate a sync site.

Isolating a protected target site

If SRDF/Star is running normally and in the STAR Protected state, the `symstar disable` command disables STAR but leaves both target sites in the Protected state, from which you can isolate either site. For example:

```
symstar -cg StarGrp disable
symstar -cg StarGrp isolate -site London
```

This action isolates site London by splitting its SRDF pairs and making the R2 devices read/write enabled to the London host.

Isolating a disconnected target site

If the site you want to isolate is in the *Disconnected* state, you must first get it to the Protected state with the `connect` and `protect` commands. For example:

```
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site London
symstar -cg StarGrp isolate -site London
```

Rejoining an isolated site

After performing testing or other tasks in London that require the isolation, you can rejoin the London site with the SRDF/Star configuration and enable SRDF/Star protection again. To do this, you must first transition London from the *Isolated* state to the *Disconnected* state. Then proceed to connect and protect. For example:

```
symstar -cg StarGrp disconnect -site London
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

After rejoining the London site, you need to reestablish any London BCV pairs that are part of the `StarGrp` composite group.

Unprotecting the target sites

To unprotect the target sites, you must first turn off SRDF/Star protection (assuming the system state is STAR Protected). The following command sequence must be executed from the workload site (`NewYork`):

```
symstar -cg StarGrp disable
symstar -cg StarGrp unprotect -site NewJersey
symstar -cg StarGrp unprotect -site London
```

Where:

- The `disable` action disables SRDF/Star protection and terminates the SDDF sessions.
- The `unprotect` action disables SRDF consistency protection and sets the STAR mode indicators.

Halting the target sites

The `halt` operation is used to prepare for a planned switch of the workload site to a target site. It suspends the SRDF links, disables all consistency protection, and sets the mode to adaptive copy disk. In addition, this operation write-disables the R1 devices and drains all invalid tracks to create a consistent copy of data at each site. To halt SRDF/Star, enter:

```
symstar -cg StarGrp halt
```

IMPORTANT

For halt operations, all RDF links between the 3 sites, including the RDF links for the recovery leg, must be online before you initiate the halt operation.

Cleanup metadata

The `symstar cleanup` command cleans up internal metadata and array cache after a failure.

The cleanup action applies only to the asynchronous site.

To clean up any internal metadata or array cache for composite group `StarGrp` remaining at the asynchronous site (`London`) after the loss of the workload site:

```
symstar -cg StarGrp cleanup -site London
```

SRDF/Star consistency group operations

This section describes the following topics:

- "Before you begin: SRDF daemon interaction"
- "Restrictions"
- "Preparing the staging area"
- "Adding devices to a concurrent SRDF/Star consistency group"
- "Adding devices to a cascaded SRDF/Star consistency group"
- "Removing devices from an SRDF/Star concurrent consistency group"
- "Removing devices from an SRDF/Star cascaded consistency group"
- "Recovering from a failed consistency group modification"

You can dynamically add or remove devices from an SRDF/Star consistency group while maintaining consistency protection if the group is in the Connected, Protected, or STAR enabled states for the following configurations:

- Concurrent SRDF/Star CG
- Concurrent SRDF/Star CG with R22 devices
- Cascaded SRDF/Star CG
- Cascaded SRDF/Star CG with R22 devices

"Consistency Group Operations" on page 151 provides additional information about managing consistency groups.

In SRDF/Star configurations, the `symstar modifycg` command with the `add` and `remove` options perform dynamic modification of SRDF/Star consistency groups.

IMPORTANT

Run the `symstar modifycg` command from the workload site.

The remove operation moves the device pairs from the SRDF/Star consistency group into the SRDF groups in the staging areas.

Before you begin: SRDF daemon interaction

Before performing any control operations on a dynamic consistency group, you must understand how the SRDF daemon (`storrdfd`) maintains consistency protection of an SRDF/Star CG during modification.

- The SRDF daemon must be running locally on the host where the `symstar modifycg` operation is issued.
- The SRDF daemon on the local host continuously monitors the SRDF/Star consistency group that is being changed.
- The SRDF daemons running on other hosts do the following:

- On hosts *not* running GNS, SRDF daemons running on Solutions Enabler versions lower than 7.3.1 stop monitoring the SRDF/Star CG during dynamic modification. These daemons see the old CG definition until the `symstar buildcg -update` command is issued.

  `symstar buildcg -update` retrieves the new SRDF/Star CG definition file from the local array and replaces the old CG definition with the updated one on that host.

- On hosts running GNS, SRDF daemons monitor the consistency group while it is being modified.

  After the SRDF/Star CG definition is modified, the GNS daemon sends the new CG definition file to all hosts local to the workload array.

  Issue the `symstar buildcg -update` command from only one host attached to each affected remote array.

  Depending on the timing of the GNS updates, there may be a brief period during which the SRDF daemon stops monitoring the SRDF/Star CG while waiting for the updated CG definition to propagate to the local GNS daemon.

  IMPORTANT

  Do not enable the `gns_remote_mirror` option in the GNS daemon's options file if you are using GNS with SRDF/Star. This option is not supported in SRDF/Star environments.

  `gns_remote_mirroring` does not remotely mirror CGs that contain concurrent or cascaded devices. If you are using GNS, enabling the gns_remote_mirror option will not mirror the CG if it includes any devices as listed in the "Mirroring exceptions" in the *EMC Solutions Enabler Array Management CLI User Guide*. Refer to the guide for a detailed description of GNS.

  To switch to a remote site, use the `symstar buildcg` command to build a definition of the CG at each site in the SRDF/Star configuration.

Restrictions

These restrictions apply to the `add` and `remove` options of the `symstar modifycg` command:

- Enginuity 5773 or higher is required.
- The `symstar modifycg` command must be executed at the workload site.
- All arrays are reachable.
- The SRDF daemon must be running locally on the host where the `symstar modifycg` command is issued.
- The `symstar modifycg` command can only move devices within one SRDF/Star triangle in the CG.
- The following options in the SRDF/Star options file must have these settings:

  SYMCLI_STAR_AUTO_DISTRIBUTE_INTERNAL_FILE=YES

SYMCLI_STAR_COMPATIBILITY_MODE =v70

- If the `symstar modifycg` command is run when one of its target sites is in the Connected state, the SRDF mode must be adaptive copy.
- When devices are virtually provisioned, all arrays in the Star triangle of the SRDF/Star CG must be running Enginuity 5874.210.168 or higher.

Note: In the event the `symstar modifycg` command fails, you can rerun the command or issue `symstar recover`. No control operations are allowed on a CG until after a recover completes on that CG.

Preparing the staging area

Before you can dynamically modify of SRDF/Star consistency groups, you must create a staging area that mirrors the configuration of the CG being used for the Star triangle that is being modified. The staging area consists of:

- SRDF groups containing the device pairs to be added to an SRDF/Star consistency group (`symstar modifycg -add` operations).
- SRDF groups for receiving the device pairs removed from an SRDF/Star consistency group (`symstar modifycg -remove` operations).
- The SRDF groups in the staging area must be established between the same arrays as the SRDF groups in the SRDF/Star consistency group being used for the Star triangle being modified.

Restrictions: SRDF/Star staging area

See also:

- "Restrictions" on page 220
- "Restrictions: SRDF groups and devices for dynamic add operations" on page 170

The following additional restrictions apply to the SRDF groups and devices in the staging area for dynamic `symstar modifycg add` operations:

- All device pairs must be set in the same mode:
  - Adaptive copy disk
  - Adaptive copy write pending for diskless R21->R2 device pairs

    Note: Adaptive copy write pending mode is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS, and diskless R21 devices are not supported on arrays running HYPERMAX OS.

- Devices in the staging area must be in one of the following SRDF pair states for each SRDF group:
  - Synchronized
  - SyncInProg with no invalid tracks
  - Suspended with no invalid tracks

If any device is Suspended on any of its SRDF groups, then all devices must be Suspended on all of their SRDF groups.

- All devices to be added in the staging area must be of the same configuration (and over the same arrays) as the Star configuration that you are updating:
  - Concurrent R1 devices
  - Cascaded R1 devices with diskless R21 devices
  - Cascaded R1 devices with non-diskless R21 devices.
- No devices in the staging area can be configured as R22 devices, but they must have an available dynamic mirror position.
- Devices in the staging area cannot be enabled for consistency protection.
- Devices in the staging area cannot be defined with SRDF/Star SDDF sessions.

Adding devices to a concurrent SRDF/Star consistency group

Use the `symstar modifycg` command to move devices between the staging area and the SRDF/Star CG, and update the CG definition.

symstar modifycg command syntax

```
symstar -cg CgName
   -i Interval
   -c Count
   -noprompt
   -v
   -sid SymmID
   -devs SymDevStart:SymDevEnd or
         SymDevName, SymDevStart:SymDevEnd or
         SymDevName... or
   -file FileName
   -stg_rdfg GrpNum,GrpNum
   -cg_rdfg CgGrpNum,CgGrpNum
   -stg_r21_rdfg R21GrpNum
   -cg_r21_rdfg CgR21GrpNum
modifycg -add [-force]
modifycg -remove
```

Where:

- `-devs SymDevStart:SymDevEnd` or `SymDevName, SymDevStart:SymDevEnd SymDevName...` or `-file FileName` - Specifies the ranges of devices to add or remove.
- `-stg_rdfg GrpNum,GrpNum` - Indicates the SRDF group(s) comprising the staging area. For a concurrent CG, two groups must be specified, separated by a comma. These SRDF groups are associated with the SRDF groups in the -cg_rdfg option. This association is based on their order in this option and -cg_rdfg.
- `-cg_rdfg CgGrpNum,CgGrpNum` - The SRDF group(s) within the SRDF/Star CG in which to add or remove devices. For a concurrent SRDF/Star CG, two SRDF groups must be specified, separated by a comma. These SRDF groups are associated with the SRDF groups in the -stg_rdfg option. This association is based on their order in this option and -stg_rdfg.

- `-stg_r21_rdfg` *`R21GrpNum`* – The SRDF group comprising the staging area at the R21 array when the configuration is cascaded. It is required for an add or remove operation when the setup is cascaded. This SRDF group is associated with the SRDF group in the -cg_r21_rdfg option.
- `-cg_r21_rdfg` *`CgR21GrpNum`* – The SRDF group connecting the R21 and R2 arrays of a cascaded SRDF/Star CG. It is only valid for operations involving cascaded R1 devices. This SRDF group is associated with the SRDF group specified in the -stg_r21_rdfg option.

Restrictions: adding devices to an SRDF/Star concurrent consistency group

The following restrictions apply to the `symstar modifycg add` operation:

- The add operation can only add new device pairs to an existing Star triangle within the SRDF/Star CG. It cannot add a new Star triangle to the SRDF/Star CG.
- If the target of the operation is a concurrent SRDF/Star CG (with or without R22 devices), the devices to be added must be concurrent R1 devices.
- If the target of the operation is a cascaded SRDF/Star CG (with or without R22 devices), the devices to be added must be cascaded R1 devices.
- If the target of the operation is a cascaded SRDF/Star CG (with or without R22 devices) and the devices to be added are cascaded R1 devices with a diskless R21, then the R21 devices in the affected triangle of the SRDF/CG CG must also be diskless.
- If the target of the operation is a cascaded SRDF/Star CG (with or without R22 devices) and the devices to be added are cascaded R1 devices with a non-diskless R21, then the R21 devices in the affected triangle of the SRDF/Star CG must also be non-diskless.

Table 31 lists the valid SRDF/Star states for adding device pairs to a CG in a concurrent SRDF/Star configuration.

Table 31 Allowable SRDF/Star states for adding device pairs to a concurrent CG

State of 1st target site (Synchronous)	State of 2nd target site (Asynchronous)	STAR state
Connected	Connected	Unprotected
Protected	Connected	Unprotected
Connected	Protected	Unprotected
Protected	Protected	Unprotected
Protected	Protected	Protected

Example The example in Figure 55 shows:

- CG `ConStarCG` spanning a concurrent SRDF/Star configuration
- The 3 arrays are: 306, 311, and 402
- The staging area contains devices 20 and 21:

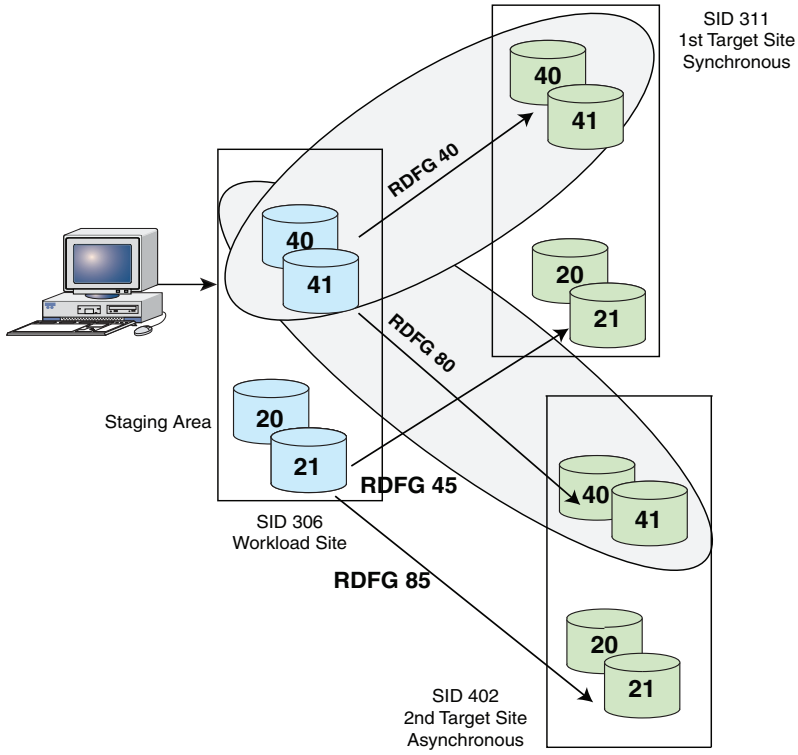


Figure 55 Adding a device to a concurrent SRDF/Star CG

To add only device 20 from the staging area into SRDF groups 40 and 80 of `ConStarCG`:

```
symstar -cg ConStarCG modifycg -add -sid 306 -stg_rdfg 45,85 -devs 20 -cg_rdfg 40,80
```

To check if device 20 was added to `ConStarCG`:

```
symstar show -cg ConStarCG -detail
```

Figure 56 shows `ConStarCg` after device 20 was added.

Note that device 21 is still in the staging area:

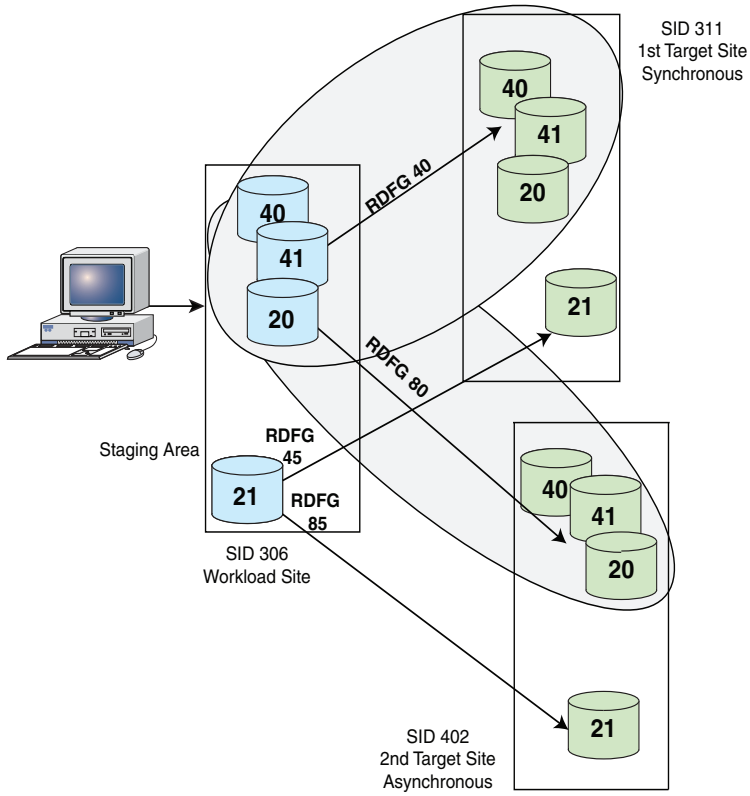


Figure 56 ConStarCG after a dynamic add operation

Adding devices to a cascaded SRDF/Star consistency group

Table 32 shows the valid states for adding device pairs to a CG in a cascaded SRDF configuration.

Table 32 **Allowable states for adding device pairs to an SRDF/Star cascaded CG**

State of 1st target site (Synchronous)	State of 2nd target site (Asynchronous)	STAR state
Connected	Connected	Unprotected
Protected	Connected	Unprotected
Protected	Protected	Unprotected
Protected	Protected	Protected

Example The example in Figure 57 shows:

- `CG CasStarCG` spanning a cascaded SRDF/Star configuration
- The 3 arrays are: 306, 311, and 402.

- The staging area contains devices 20 and 21:

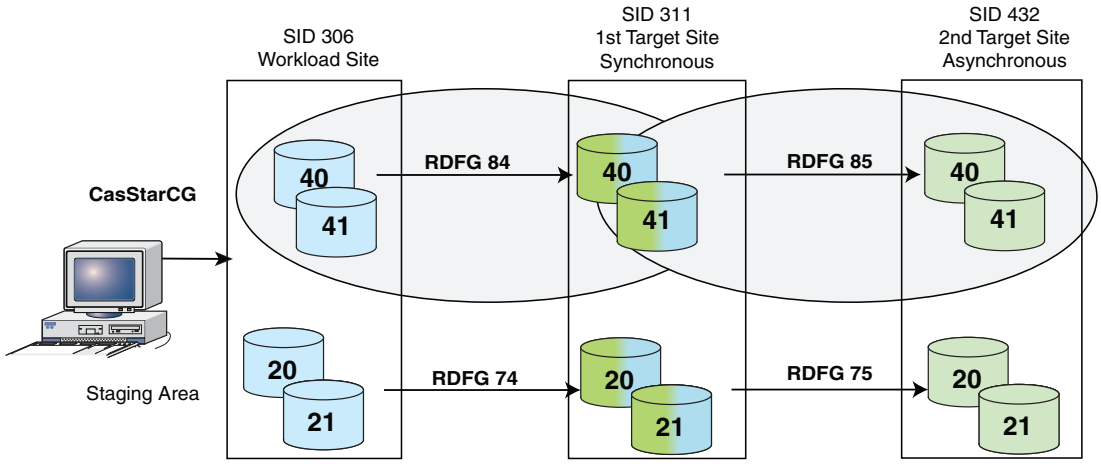


Figure 57 Adding devices to a cascaded SRDF/Star CG

1. Use the `symstar -cg GroupName modifycg -add` command to move the devices from the staging area to the SRDF group(s).

   To move devices 20 and 21 from the staging area to SRDF groups 84 and 85 of `CasStarCG`:

```
symstar -cg CasStarCG modifycg -add -sid 306 -stg_rdfg 74 -devs 20:21 -stg_r21_rdfg 75 -cg_rdfg
84 -cg_r21_rdfg 85
```

2. Use the `symstar show -cg GroupName -detail` command to verify that the devices were moved.

   To verify devices 20 and 21 were added to `CasStarCG`:

```
symstar show -cg CasStarCG -detail
```

Figure 58 shows the configuration after the move:

- Devices 20 and 21 were added to `CasStarCG`.
- The staging area contains empty SRDF groups 74 and 75:

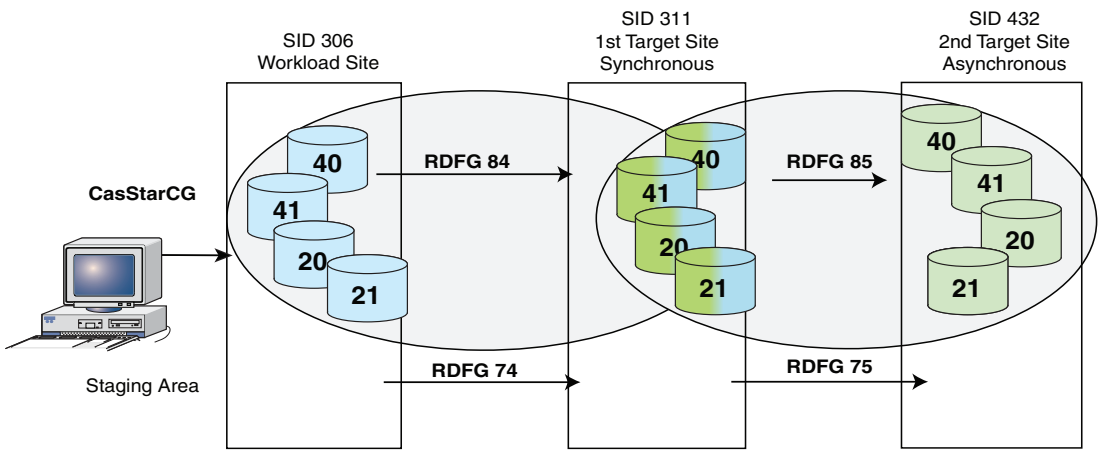


Figure 58 CasStarCG after a dynamic add operation

Pair states of devices in an SRDF/Star CG after a symstar modifycg -add

Table 33 shows the pair states of the devices in the SRDF/Star CG after the `symstar modifycg -add` command completes. These pair states are based on the state of the SRDF/Star site and the SRDF mode of the device pairs in the CG.

Table 33 Pair states of the SRDF devices after symstar modifycg -add completion

State of SRDF/Star sites	Mode of device pairs in CG	Pair state of devices in CG after symstar modifycg -add	Possible delay for symstar modifycg -add command
Connected	Adaptive copy disk	Synchronized or SyncInProg	No delay because command completes when pair is SyncInProg.
Protected	SRDF/S	Synchronized	Completes when devices are synchronized.
	SRDF/A	Consistent without invalid tracks	Completes when the consistency exempt option (`-cons_exempt`) clears on the devices added to the CG.
Star Protected	SRDF/S	Synchronized	Completes when devices are synchronized.
	SRDF/A	Consistent without invalid tracks	Completes when devices are recoverable.

Removing devices from consistency groups

The dynamic modifycg remove operation moves the device pairs from an SRDF/Star consistency group to the staging area. If the SRDF/Star CG has R22 devices, a deletepair operation on the recovery links of the CG is performed automatically.

Note: Never use the dynamic modifycg remove operation to remove an existing triangle from the SRDF/Star CG.

Restrictions: SRDF/Star groups and devices for dynamic remove operations

The following restrictions apply to the SRDF groups and devices in the staging area for dynamic `symstar modifycg remove` operations:

- SRDF groups in the staging area are not in the STAR state.
- SRDF groups in the staging area are not in asynchronous mode.

Removing devices from an SRDF/Star concurrent consistency group

Table 34 shows the valid states for removing device pairs from a CG in a concurrent SRDF/Star configuration.

Table 34 Allowable states for removing device pairs from an SRDF/Star concurrent CG

State of 1st target site (Synchronous)	State of 2nd target site (Asynchronous)	Star state
Connected	Connected	Unprotected
Protected	Connected	Unprotected
Connected	Protected	Unprotected
Protected	Protected	Unprotected
Protected	Protected	Protected

Example To move device 35 from the RDG groups 40 and 80 of `ConStarCG` into SRDF groups 45 and 85 of the staging area:

```
symstar -cg ConStarCG modifycg -remove -sid 306 -stg_rdfg 45,85 -devs 35 -cg_rdfg 40,80
```

To check if the dynamic remove operation was successful for `ConStarCG`:

```
symstar show -cg ConStarCG -detail
```

Removing devices from an SRDF/Star cascaded consistency group

Table 35 shows the valid states for removing device pairs from a CG in a cascaded SRDF configuration.

Table 35 Allowable SRDF/Star states for removing device pairs from a cascaded CG

State of 1st target site (Synchronous)	State of 2nd target site (Asynchronous)	Star state
Connected	Connected	Unprotected
Protected	Connected	Unprotected
Protected	Protected	Unprotected
Protected	Protected	Protected

Example To move devices 21 and 22 from SRDF groups 84 and 85 of `ConStarCG` into SRDF groups 74 and 75 of the staging area:

```
symstar -cg ConStarCG modifycg -remove -sid 306 -stg_rdfg 74 -devs 21:22 -stg_r21_rdfg 75
-cg_rdfg 84 -cg_r21_rdfg 85
```

To check if the dynamic remove operation was successful:

```
symstar -cg ConStarCG show -detail
```

Recovering from a failed consistency group modification

Details about change operations (target CG, SRDF groups, staging area, and operation type) are stored in the SFS.

If a modifycg operation fails and all SRDF/Star sites are reachable:

1. Reissue the modifycg command using *exactly* parameters as the command that failed.
2. If the command fails again, execute the following command at the workload site:

   symstar -cg *CGName* recover

   If the workload site or any of the SRDF/Star CG sites are unreachable, use `-force`:

   symstar -cg *CGName* recover -force

The `symstar recover` command uses all existing information of a dynamic modifycg operation in SFS.

The recover operation either completes the unfinished steps of the dynamic modifycg operation or rolls back any tasks performed on the CG by this operation, placing the CG into its original state before failure.

Example In this example, re-try of the `symstar modifycg add` operation run from Site A fails due to a trip event at Site C:

1. From Site A, use the `symstar -cg GroupName query -detail` command to display whether the Composite Group State is RecoveryRequired.

   To display CG SampleCG:

   symstar -cg SampleGCG query -detail

2. Use the `symstar -cg GroupName recover -force` command to re-try the failed operation.

   To re-try the failed `symstar modifycg add` for CG SampleCG:

   symstar -cg SampleCG recover -force

   Output varies depending on whether the recovery succeeds.

   If the recovery succeeds, final line of output:

   RecoverAdd.....Done.

   If the recovery fails, SRDF rolls back the operation and removes any devices added before the failure. Final line of output:

   RecoverRollBack.....Done.

SRDF pair states of devices in an SRDF/Star CG after a recovery

Table 36 shows the possible pair state of the devices in the SRDF/Star CG after the symstar recover operation completes.

The synchronous target site and/or the asynchronous target site can be in the Disconnected or PathFail state when the recover operation is issued for a concurrent SRDF/Star CG or a cascaded SRDF/Star CG.

Table 36 Possible pair states of the SRDF devices after a recovery

State of SRDF/Star sites	Mode of device pairs in CG	Pair state of devices in CG after a recovery
Disconnected	Adaptive copy disk	Suspended[1]
PathFail	SRDF/S	Suspended[1]
	SRDF/A	Suspended[1]

1. The SRDF pair state can be Partitioned if the SRDF link is offline.

Command failure while in the Connected state

While in the SRDF/Star Connected state, if a dynamic modification operation fails and indicates the SRDF mode of one or more legs in the STAR CG is invalid, issue the `symstar configure -reset rdf_mode` command at the workload site. This command resets the device pairs in the SRDF/Star CG to adaptive copy mode. After the `symstar configure -reset rdf_mode` successfully completes, you can reissue the `symstar modifycg` operation.

Recovery and switch operations: concurrent SRDF/Star

This section describes the following topics:

- "Recovery operations: concurrent SRDF/Star"
- "Workload switching: concurrent SRDF/Star"

Recovery operations: concurrent SRDF/Star

This section describes the following topics:

- "Recovering from transient faults: concurrent SRDF/Star"
- "Recovery from a transient fault without reconfiguration: concurrent SRDF/Star"
- "Recovery from transient fault with reconfiguration: concurrent SRDF/Star"

Recovering from transient faults: concurrent SRDF/Star

A transient fault does not disrupt the production workload site. Only the transfer of data across the link is affected. Transient faults during normal SRDF/Star operations require a recovery action.

An SRDF/Star fault caused by network or remote storage controller faults is a transient fault.

This section describes recovery when a transient fault occurs while SRDF/Star is in the *Protected* or *Star_Protected* states.

If a transient fault occurs on a link that is in the *Connected* state, the link is disconnected. Restarting synchronization again from a *Disconnected* state (after correcting the cause of the failure) requires only the `connect` action.

Figure 59 shows a temporary interruption on the SRDF/A link in a concurrent SRDF/Star environment:

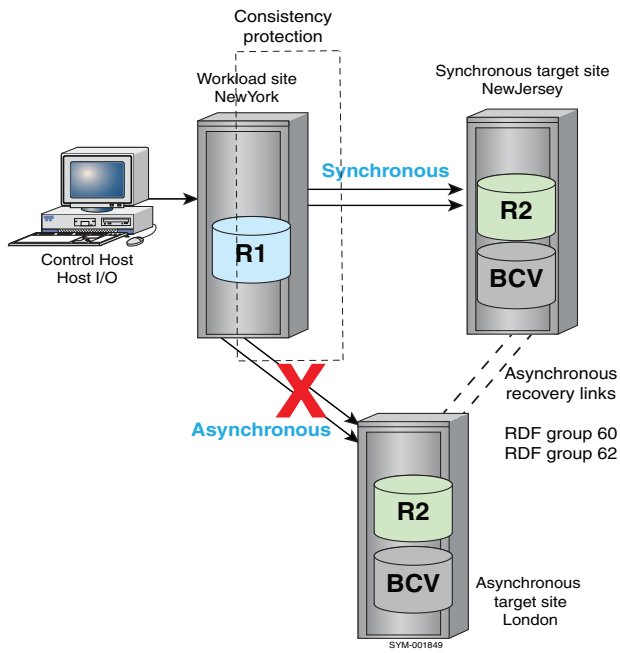


Figure 59 Transient failure: concurrent SRDF/Star

There are two methods to cleanup and restore SRDF/Star:

- When the transient fault is corrected, you can clean up the internal metadata and the cache at the asynchronous target site and return the site to SRDF/Star Protected. "Recovery from a transient fault without reconfiguration: concurrent SRDF/Star" on page 265 describes the steps to recover from a transient fault on the SRDF/A link when the fault has been repaired.
- If you cannot wait for the transient fault to be corrected, you can reconfigure SRDF/Star to recover the asynchronous site. "Recovery from transient fault with reconfiguration: concurrent SRDF/Star" on page 266 describes the steps to avoid a long wait when the asynchronous site must be recovered sooner than the transient fault will be repaired.

Recovery from a transient fault without reconfiguration: concurrent SRDF/Star

If the synchronous target (NewJersey in Figure 59) state is Protected, and the asynchronous target (London) state is PathFail.

1. Use the `symstar -cg Groupname reset` command to clean up any internal metadata or cache remaining at the asynchronous site after the transient fault occurred.

   To cleanup cache and metatdata for CG StarGrp at site London:

   ```
   symstar -cg StarGrp reset -site London
   ```

   NOTICE

   If remote BCVs are configured, split the remote BCVs after a transient fault to maintain a consistent image of the data at the remote site until it is safe to reestablish the BCVs with the R2 devices.

Resynchronization temporarily compromises the consistency of the R2 data until the resynchronization is fully completed. The split BCVs retain a consistent restartable image of the data volumes during periods of SRDF/Star resynchronization.

The next step varies depending on whether you are protecting SRDF/Star data at the remote site with TimeFinder BCVs:

- If you are protecting SRDF/Star data at the remote site with TimeFinder BCVs, proceed to 2.
- If you are not protecting SRDF/Star data at the remote site with TimeFinder BCVs, skip to Step 3.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the remote site, perform the appropriate TimeFinder actions.

   To split off a consistent restartable image of the data volumes prior to resynchronization at the asynchronous target (London) site:

   ```
   symmir -cg StarGrp split -star -rdf -rdfg name:London
   ```

3. Use the `symstar -cg GroupName` command with the `connect, protect,` and `enable` options to return the asynchronous site to the SRDF/Star configuration.

   To connect, protect and enable the CG StarGrp at the London site:

   ```
   symstar -cg StarGrp connect -site London
   symstar -cg StarGrp protect -site London
   symstar -cg StarGrp enable
   ```

4. If any London BCV pairs are part of the composite group, use the `symmir -cg GroupName establish` command to reestablish them.

   To reestablish the BCV pairs

   ```
   symmir -cg StarGrp establish -star -rdf -rdfg name:London
   ```

Recovery from transient fault with reconfiguration: concurrent SRDF/Star

If the transient fault persists, you may not want to wait for the fault to be repaired to reestablish your SRDF/Star protection.

This procedure describes the steps to recover your SRDF/Star by reconfiguring the path between the synchronous site and the asynchronous site.

This alternate method avoids a long wait when the asynchronous site needs to be recovered sooner than the transient fault will be repaired.

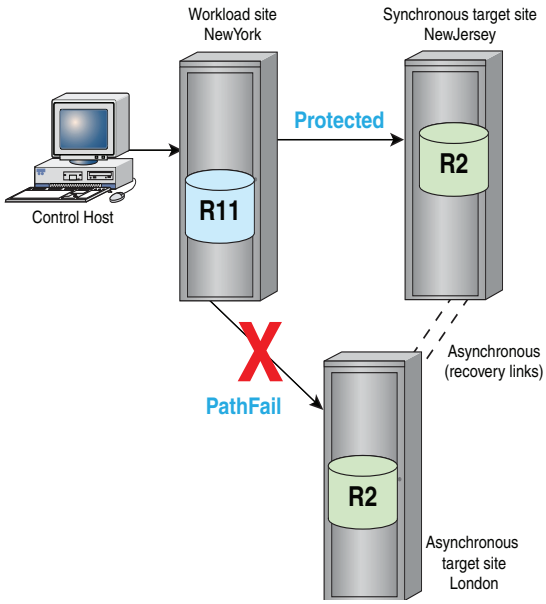


Figure 60 Transient fault recovery: before reconfiguration

Figure 60 shows a fault where the links between the workload site and the asynchronous target sites are lost.

- The asynchronous target site (London) is accessible by the recovery SRDF groups at the synchronous site (NewJersey).
- The failure causes SRDF/Star to enter a tripped state.

You can restore SRDF/Star protection to the asynchronous target site by reconfiguring from concurrent SRDF/Star to cascaded mode.

Use the `reconfigure` operation (to change the mode to cascade) as the initial recovery step.

symstar reconfigure command syntax

```
symstar -cg CgName [-noprompt] [-i Interval][-c Count]
   -wkload SiteName
   -opmode concurrent | cascaded
   reconfigure
      -path SrcSiteName:TgtSiteName
      -site TgtSiteName
      -remove SrcSiteName:TgtSiteName
      -full
      -reset
      -force
```

Where:

- `-path SrcSiteName:TgtSiteName` specifies the sites on which the new SRDF pairs are created when the reconfigure command is issued.
- `-site TgtSiteName` specifies the SiteName to apply the given action.
- `-reset` Performs a reset action on the path when the reconfigure action is issued.
- `-remove SrcSiteName:TgtSiteName` Specifies the sites on which the SRDF pairs are removed.

Usage notes

- If the asynchronous target site is in the Disconnected state and STAR is unprotected, you must use the `-full` option.
- If the asynchronous target site is in the PathFail state and STAR is unprotected, use the `-reset` and `-full` options.
- Use the `-full` option only when an SRDF incremental resync is not available.
- Use the `recover` operation to recover from PathFail (asynchronous target site) and a tripped state (SRDF/Star)

Example To reconfigure CG StarGrp so that the path to London is NewJersey -> London:

```
symstar -cg StarGrp reconfigure -reset -site London -path NewJersey:London
```

The topology of the configuration is now cascaded:

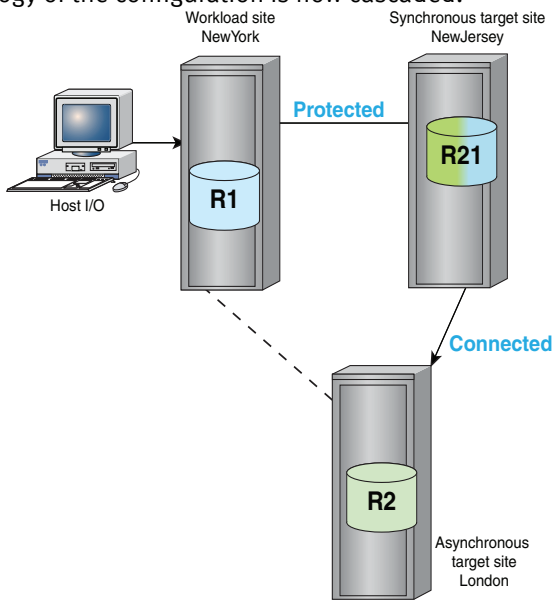


Figure 61 Transient fault recovery: after reconfiguration

Workload switching: concurrent SRDF/Star

This section describes the following topics:

- "Planned workload switching: concurrent SRDF/Star"
- "Unplanned workload switching: concurrent SRDF/Star"
- "Unplanned workload switch to synchronous target site: concurrent SRDF/Star"
- "Unplanned workload switch to asynchronous target site: concurrent SRDF/Star"
- "Switch back to the original workload site: concurrent SRDF/Star"

Planned workload switching: concurrent SRDF/Star

A planned workload switch operation switches the workload function to one of the remote target sites, even when:

- The original workload site is operating normally,

- The system state is STAR Protected, or
- The target sites are at least Connected.

IMPORTANT

For planned switch operations, all RDF links between the 3 sites, including the RDF links for the recovery leg, must be online before you initiate the planned switch operation.

To switch the workload from the original site:

1. Use the `symstar query` command to confirm the system state.
2. Stop the application workload at the current workload site, unmount the file systems, and export the volume groups.
3. Use the SRDF/Star `halt` action from the control host.

   To halt the CG StarGrp:

   ```
   symstar -cg StarGrp halt
   ```

   Note: If you change your mind after halting SRDF/Star, issue the `halt -reset` command to restart the workload site on the same host.

   The `halt` action at the initial workload site (`NewYork`):

   - Disables the R1 devices
   - Waits for all invalid tracks and cycles to drain,
   - Suspends the SRDF links,
   - Disables SRDF consistency protection, and
   - Sets the STAR mode indicators.

The target sites transition to the Halted state, with all three sites having the data, as shown in Figure 62:

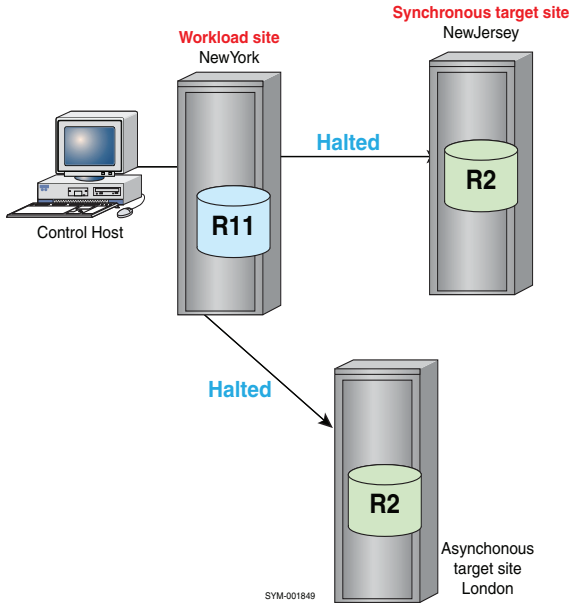


Figure 62 Concurrent SRDF/Star: halted

4. From a control host at the synchronous target site (`NewJersey`), use the switch command to switch the workload to the synchronous target site (`NewJersey`).

   ```
   symstar -cg StarGrp switch -site NewJersey
   ```

   The resulting SRDF/Star state is shown in Figure 63:

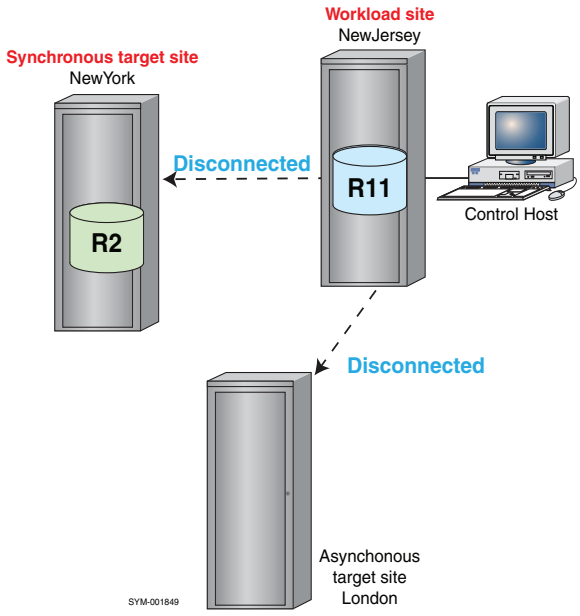


Figure 63 Concurrent SRDF/Star: switched

5. From a control host at the synchronous target site (`NewJersey`), use two `connect` commands to

- Connect `NewJersey` to `NewYork` (synchronously)
- Connect `NewJersey` to `London` (asynchronously):

```
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp connect -site London
```

The resulting SRDF/Star state is shown in Figure 64:

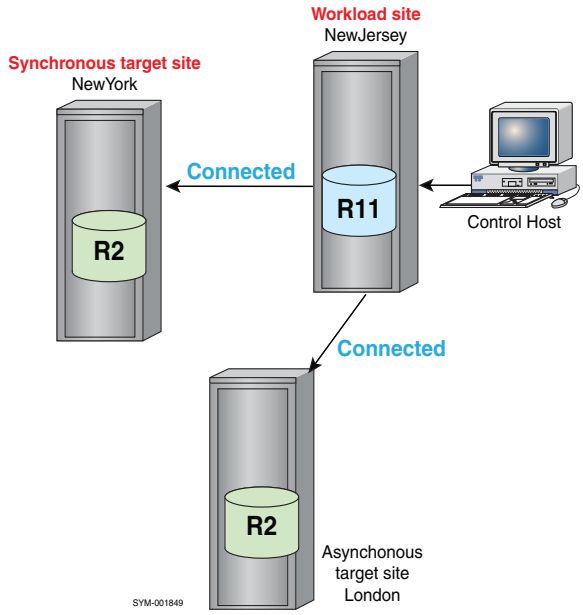


Figure 64 Concurrent SRDF/Star: connected

6. From a control host at the synchronous target site (`NewJersey`), use two `protect` commands and the `enable` command to:
   - Protect `NewJersey` to `NewYork`
   - Protect `NewJersey` to `London`
   - Enable SRDF/Star

```
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The resulting SRDF/Star state is shown in Figure 65:

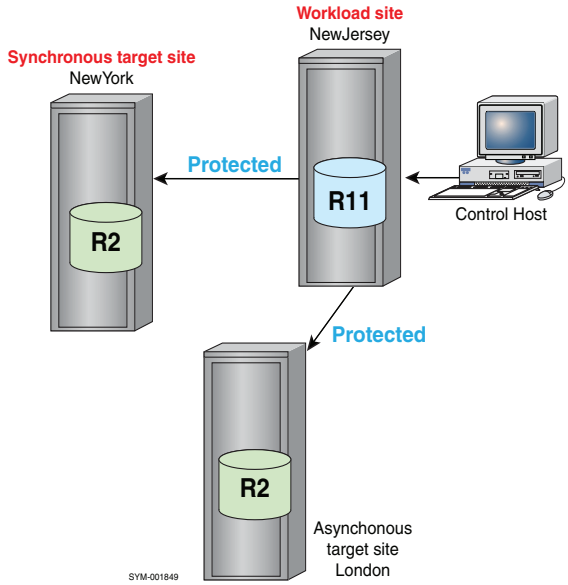


Figure 65 Concurrent SRDF/Star: protected

Unplanned workload switching: concurrent SRDF/Star

Loss of the workload site (`NewYork`) is a disaster because it disrupts the workload.

Use the `switch` command to:

- Switch the workload to either one of the remote sites, and
- Resume data replication

You can switch the workload to either the synchronous or asynchronous target site.

If the loss of the workload site was caused by a rolling disaster, the data at the synchronous target site can be ahead of the data at asynchronous site, or vice versa.

You can specify which site's data to keep.

Figure 66 shows concurrent SRDF/Star where a disaster fault has caused the loss of the workload site (`NewYork`):

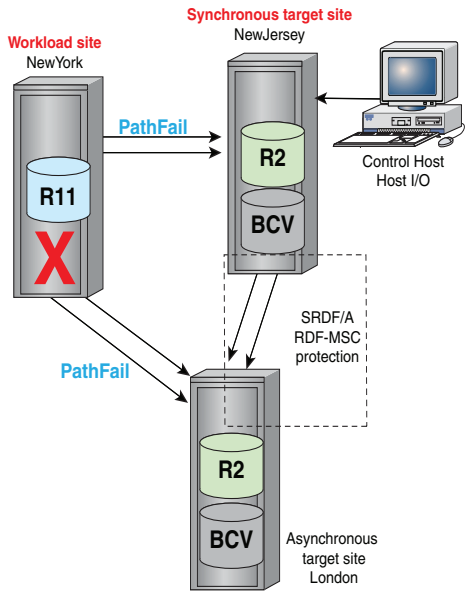


Figure 66 Loss of workload site: concurrent SRDF/Star

Unplanned workload switch to synchronous target site: concurrent SRDF/Star

In the following example, loss of the workload site (`NewYork`) has resulted in a system state of *NewJersey:Pathfail*, *London:Pathfail*, and *STAR:Tripped*.

Note: If you switch the workload to the synchronous target site but choose to keep the data from the asynchronous target site, there is a wait for all the SRDF data to synchronize before the application workload can be started at the synchronous site.

The `symstar switch` command does not return control until the data is synchronized.

This procedure:

- Brings up the synchronous `NewJersey` site as the new workload site.
- Asynchronously replicates data from `NewJersey` data to the asynchronous target site (`London`).

Note: If the links from the workload to the asynchronous target are in the *TransmitIdle* state, use the following command to get the asynchronous site to the *PathFail* state:

```
symstar -cg StarGrp disconnect -trip -site London
```

1. From a control host at the synchronous target site (`NewJersey`), use the `symstar cleanup` command to clean up any internal metadata or cache remaining at the asynchronous site.

   To clean up the London site:

   ```
   symstar -cg StarGrp cleanup -site London
   ```

Note: After a workload site failure, splitting the remote BCVs maintains a consistent image of the data at the remote site until it is safe to reestablish the BCVs with the R2 devices.

The next step varies depending on whether you are protecting SRDF/Star data at the remote site with TimeFinder BCVs:

- If you are protecting SRDF/Star data at the remote site with TimeFinder BCVs, proceed to 2.
- If you are not protecting SRDF/Star data at the remote site with TimeFinder BCVs, skip to Step 3.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the `London` site, perform the appropriate TimeFinder actions.

   Prior to the switch and resynchronization between NewJersey and London, there is no existing SRDF relationship between the synchronous and asynchronous target sites.

   BCV control operation must be performed with a separate device file instead of the composite group.

   In the following example, the device file (`StarFileLondon`) defines the BCV pairs on array 13 in London.

   To split off a consistent restartable image of the data volumes during the resynchronization process using the device file:

   ```
   symmir -f StarFileLondon split -star -sid 13
   ```

3. From a control host at the synchronous target site (`NewJersey`), use the `symstar switch` command to start the workload at the specified site. The following command:

   - Specifies `NewJersey` as the new workload site (`-site NewJersey`)
   - Retains the data at the `NewJersey` data instead of the `London` data (`-keep_data NewJersey`):

   ```
   symstar -cg StarGrp switch -site NewJersey -keep_data NewJersey
   ```

   The resulting SRDF/Star state is shown in Figure 67:

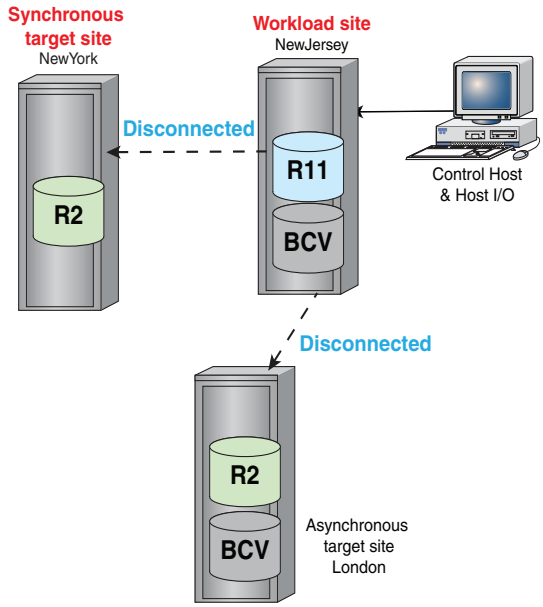


Figure 67 Workload switched to synchronous site: concurrent SRDF/Star

4. From a control host at the synchronous target site (`NewJersey`), use the `connect` command to connect `NewJersey` to `London` (asynchronously):

   ```
   symstar -cg StarGrp connect -site London
   ```

   The resulting SRDF/Star state is shown in Figure 68:

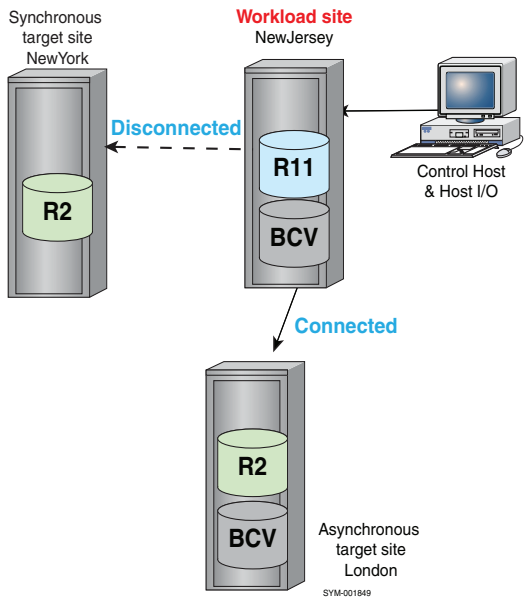


Figure 68 Concurrent SRDF/Star: new workload site connected to asynch site

5. From a control host at the synchronous target site (`NewJersey`), use the `protect` and the `enable` commands to:

   - Protect `NewJersey` to `London`
   - Enable SRDF/Star

```
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The resulting SRDF/Star state is shown in Figure 69:

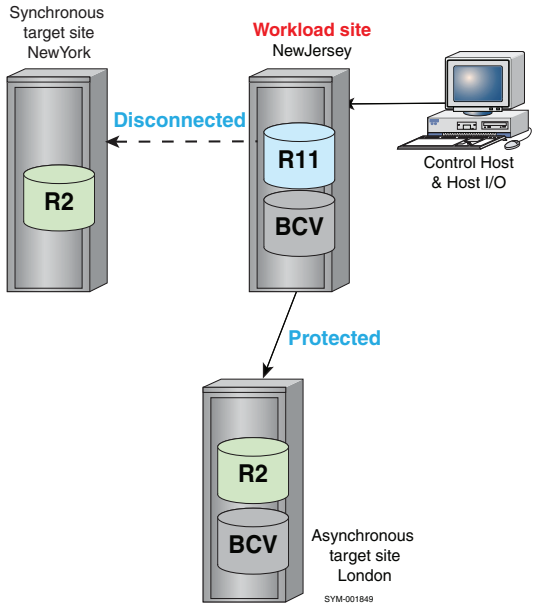


Figure 69 Concurrent SRDF/Star: protected to asynchronous site

The `connect` and `protect` actions:

- Reconfigure the SRDF devices between `NewJersey` and `London` into SRDF pairs with R1 devices at site NewJersey paired with the R2 devices at site London.
- Perform the differential resynchronization of the data between NewJersey and London.

When the recovery tasks are complete, the `NewJersey` workload is remotely protected through an asynchronous link to `London`.

NOTICE

You can begin the workload at `NewJersey` any time after the `switch` action completes. However, if you start the workload before completing the `connect` and `protect` actions, you will have no remote protection until those actions complete.

The next step varies depending on whether you are protecting SRDF/Star data at the remote site with TimeFinder BCVs:

- If you are protecting SRDF/Star data at the remote site with TimeFinder BCVs, proceed to 6.
- If you are not protecting SRDF/Star data at the remote site with TimeFinder BCVs, skip to Step 7.

6. Reestablish any BCV pairs at the `London` site. Use either:
   - The device file syntax (`-f StarFileLondon`) or,
   - The `-cg` syntax (if you have associated the `London` BCV pairs with the `StarGrp` composite group on the control host).

To reestablish `London` BCV pairs in the composite group `StarGrp` using the `-cg` syntax:

```
symmir -cg StarGrp establish -star -rdf -rdfg name:London
```

7. When the `NewYork` site is repaired, you may want to bring `NewYork` back into the SRDF/Star while retaining the workload site at `NewJersey`.

   For example, to recover and enable the `NewYork` site, enter the following commands from the `NewJersey` control host:

```
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp enable
```

   The resulting SRDF/Star state is shown in Figure 70:

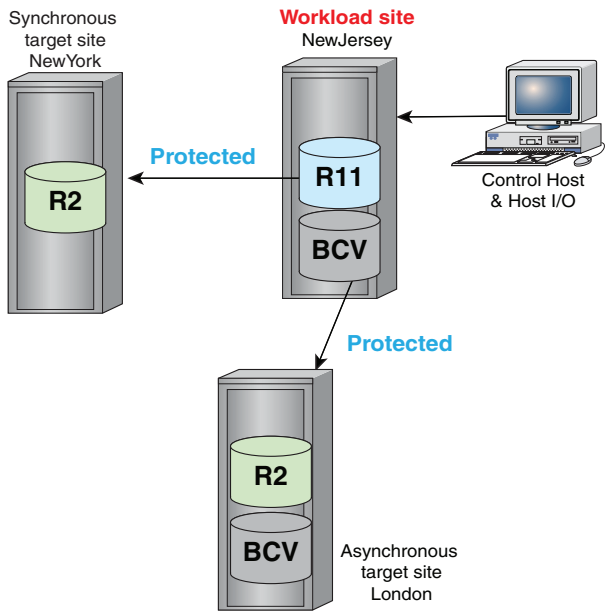


Figure 70 Concurrent SRDF/Star: protect to all sites

Unplanned workload switch to asynchronous target site: concurrent SRDF/Star

In the following example, loss of the workload site (`NewYork`) has resulted in a system state of *NewJersey:Pathfail*, *London:Pathfail*, and *STAR:Tripped*.

Note: If you switch the workload to the asynchronous target site but choose to keep the data from the synchronous target site, there is a wait for all the SRDF data to synchronize before the application workload can be started at the asynchronous site.

The `symstar switch` command does not return control until the data is synchronized.

This procedure:

- Brings up the asynchronous `London` site as the new workload site.
- Asynchronously replicates data from `London` data to the asynchronous target site (`NewJersey`).

1. From a control host at the asynchronous target site (`London`), use the `symstar cleanup` command to clean up any internal metadata or cache remaining at the asynchronous site.

   To clean up the London site:

   ```
   symstar -cg StarGrp cleanup -site London
   ```

   Note: After a workload site failure, splitting the remote BCVs maintains a consistent image of the data at the remote site until it is safe to reestablish the BCVs with the R2 devices.

   The next step varies depending on whether you are protecting SRDF/Star data at the remote site with TimeFinder BCVs:

   - If you are protecting SRDF/Star data at the remote site with TimeFinder BCVs, proceed to 2.
   - If you are not protecting SRDF/Star data at the remote site with TimeFinder BCVs, skip to Step 3.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the `NewJersey` site, perform the appropriate TimeFinder actions.

   Prior to the switch and resynchronization between NewJersey and London, there is no existing SRDF relationship between the synchronous and asynchronous target sites.

   BCV control operation must be performed with a separate device file instead of the composite group.

   In the following example, the device file (`StarFileNewJersey`) defines the BCV pairs on array 13 in London.

   To split off a consistent restartable image of the data volumes during the resynchronization process using the device file:

   ```
   symmir -f StarFileNewJersey split -star -sid 16
   ```

3. From a control host at the asynchronous target site (`London`), use the `symstar switch` command to start the workload at the specified site. The following command:

   - Specifies `London` as the new workload site (`-site NewJersey`)
   - Retains the data at the `NewJersey` data instead of the `London` data (`-keep_data NewJersey`):

   ```
   symstar -cg StarGrp switch -site London -keep_data NewJersey
   ```

   The workload site switches to London and the R2 devices at London become R1 devices.

   The London site connects to the NewJersey site and retrieves the NewJersey data.

   Note: The `connect` action is not required because the `switch` action specified that SRDF retrieve the remote data from the NewJersey site.

   The resulting SRDF/Star state is shown in Figure 71:

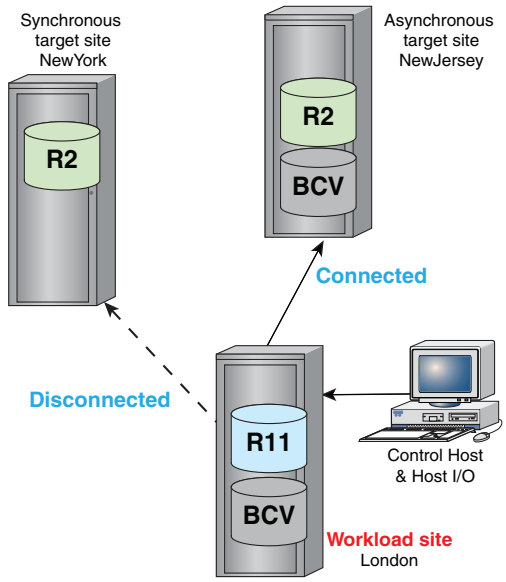


Figure 71 Workload switched to asynchronous site: concurrent SRDF/Star

4. From a control host at the asynchronous target site (`London`), use the `protect` command to protect `London` to `NewJersey`

   ```
   symstar -cg StarGrp protect -site NewJersey
   ```

   The resulting SRDF/Star state is shown in Figure 69:

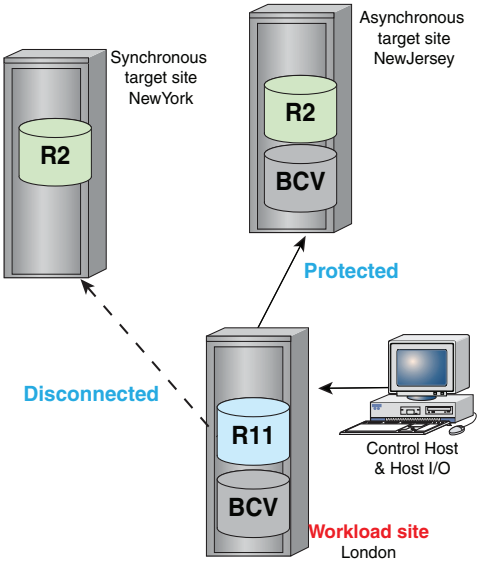


Figure 72 Concurrent SRDF/Star: protected to asynchronous site

NOTICE

London is now using the `NewJersey` data. You cannot start the application workload in `London` until the `switch` action completes. This ensures that all of the SRDF pairs are synchronized prior to starting the workload.

The `symstar switch` command blocks other action until it completes.

The next step varies depending on whether you are protecting SRDF/Star data at the remote site with TimeFinder BCVs:

- If you are protecting SRDF/Star data at the remote site with TimeFinder BCVs, proceed to 5.
- If you are not protecting SRDF/Star data at the remote site with TimeFinder BCVs, skip to Step 6.

5. Reestablish any BCV pairs at the `NewJersey` site. Use either:
   - The device file syntax (`-f StarFileNewJersey`) or,
   - The `-cg` syntax (if you have associated the `NewJersey` BCV pairs with the `StarGrp` composite group on the control host).

   To reestablish `NewJersey` BCV pairs in the composite group `StarGrp` using the `-cg` syntax:

   `symmir -cg StarGrp establish -star -rdf -rdfg name:NewJersey`

6. The London site is at asynchronous distance from both NewYork and NewJersey. SRDF/Star supports only one asynchronous site.

   When the `NewYork` site is repaired, you cannot connect *and* protect NewYork without switching the workload back to a configuration that has only one asynchronous site (`NewYork` or `NewJersey`).

   However, you can *connect* to NewYork. The `connect` action sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links.

   To connect to NewYork, use the `connect` command from the London site:

   `symstar -cg StarGrp connect -site NewYork`

   The resulting SRDF/Star state is shown in Figure 69:

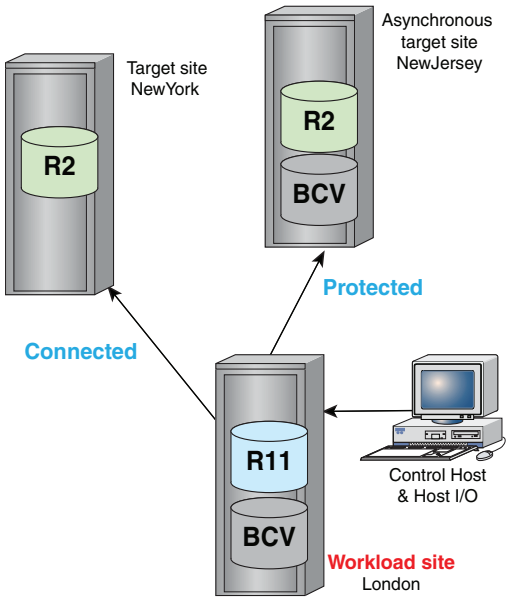


Figure 73 Concurrent SRDF/Star: one asynchronous site not protected

If the workload remains at the asynchronous `London` site, you can perform a `protect` action on `NewYork` only if you first unprotect `NewJersey`.

The `protect` action transitions the link from adaptive copy mode to asynchronous mode and enables SRDF consistency protection.

The `symstar enable` action is blocked because there is already one asynchronous link in the Star.

Note: *Using SYMCLI to Implement SRDF/Star* Technical Note provides expanded operational examples for SRDF/Star.

Switch back to the original workload site: concurrent SRDF/Star

When the original workload site returns to normal operations, you can switch back to the original workload site to reestablish the original SRDF/Star configuration.

To switch back to the original workload site:

- You must be able to completely synchronize the data at all three sites.
- The current workload site's SRDF links must be connected to the other two sites.

The states that allow you switch back to the original workload site vary depending on whether the workload was switched to the synchronous target site or the asynchronous target site:

- If you switched the workload to the synchronous target site, the following states are required to switch back:
  - STAR Protected
  - Both target sites are *Protected*
  - One target site is *Protected* and the other is *Connected*
  - Both target sites are *Connected*

- If you switched the workload to the asynchronous target site, the following states are required to switch back:
  - One target site is Protected and the other is Connected
  - Both target sites are Connected

The following procedure assumes the original workload site is NewYork, but the workload is now running at the synchronous site NewJersey. This configuration is depicted in Figure 70 on page 277.

1. Stop the workload at the site where your control host is connected.
2. Use the `halt` command from the control host where the workload is running.

   To halt SRDF from the NewJersey control host:

   ```
   symstar -cg StarGrp halt
   ```

   The halt action:

   - Disables the R1 devices,
   - Waits for all invalid tracks and cycles to drain,
   - Suspends the SRDF links,
   - Disables SRDF consistency protection, and
   - Sets the STAR indicators.

   The target sites transition to the *Halted* state, and all the data on all three sites is the same.
3. Run the following commands from the control host at the original site of the workload (*NewYork*):

   ```
   symstar -cg StarGrp switch -site NewYork
   symstar -cg StarGrp connect -site NewJersey
   symstar -cg StarGrp connect -site London
   symstar -cg StarGrp protect -site NewJersey
   symstar -cg StarGrp protect -site London
   symstar -cg StarGrp enable
   ```

- The workload is switched to NewYork, and
- NewYork is (synchronously) connected to NewJersey
- NewYork is (asynchronously) connected to London
- The state is *STAR Protected*.

Recovery and switch operations: cascaded SRDF/Star

This section describes the following topics:

- "Recovery operations: cascaded SRDF/Star"
- "Workload switching: cascaded SRDF/Star"

Recovery operations: cascaded SRDF/Star

This section describes the following topics:

- "Recovering from transient faults without reconfiguration: cascaded SRDF/Star"
- "Recovering from transient faults with reconfiguration: cascaded SRDF/Star"

Recovering from transient faults: cascaded SRDF/Star

Figure 74 shows a temporary interruption (transient fault) on the SRDF/A link in a cascaded SRDF/Star environment:

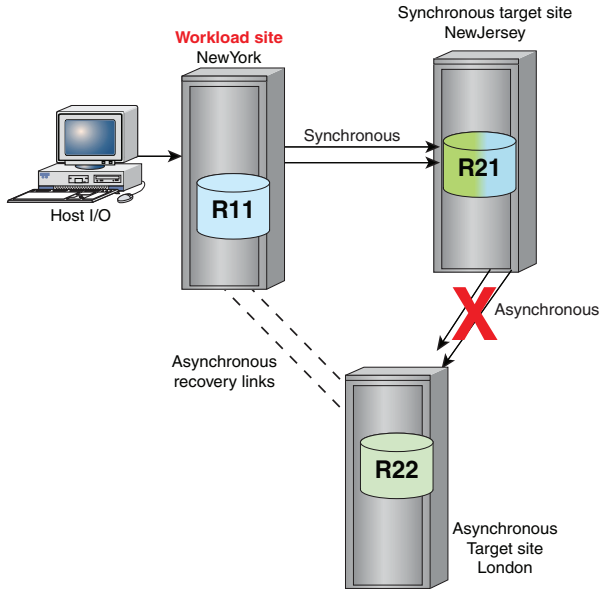


Figure 74 transient failure: cascaded SRDF/Star

There are two methods to cleanup and restore SRDF/Star:

- When the transient fault is corrected, you can clean up the internal metadata and the array cache at the asynchronous target site and return the site to SRDF/Star Protected. "Recovering from transient faults without reconfiguration: cascaded SRDF/Star" on page 283 describes the steps to recover from a transient fault on the SRDF/A link when the fault has been repaired.
- If you cannot wait for the transient fault to be corrected, you can reconfigure SRDF/Star to recover the asynchronous site. "Recovering from transient faults with reconfiguration: cascaded SRDF/Star" on page 285 describes the steps to avoid a long wait when the asynchronous site must be recovered sooner than the transient fault will be repaired.

Recovering from transient faults without reconfiguration: cascaded SRDF/Star

Figure 75 shows the SRDF states when links to the asynchronous target site are down:

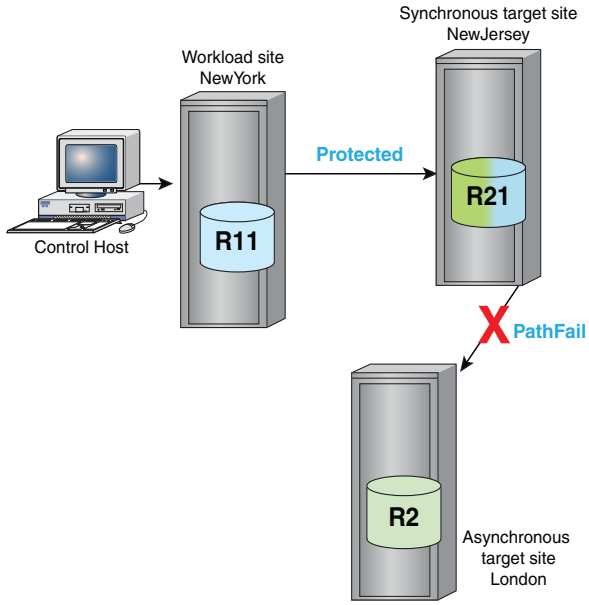


Figure 75 Cascaded SRDF/Star with transient fault

The SRDF devices are now in the Suspended state.

1. Use the `symrdf list` command to display the state the state of SRDF devices and the SRDF links that connect them.

   See Table 6, "Options for symrdf list command," for a list of symrdf list command options.

   The next step varies depending on the state of the links to the asynchronous target site (London).

   - If the links to the asynchronous target are in the *TransmitIdle* state, proceed to Step 2.
   - If the links to the asynchronous target are in the *PathFail* state, skip to Step 3.

2. Use the `symstar -cg GroupName disconnect -trip` command to transition links to the asynchronous site to the *PathFail* state.

   ```
   symstar -cg StarGrp disconnect -trip -site London
   ```

3. Use the `symrdf list` command to verify the configuration is now has the following states:

   Synchronous target site (NewJersey): *Protected*

   Asynchronous target site (London): *PathFail*

   STAR state: *Tripped*

4. From the control host at the workload site, use the `symstar -cg GroupName reset` command to clean up any internal metadata or cache remaining at the asynchronous site after the transient fault occurred.

   To cleanup cache and metadata for CG StarGrp at site London:

   ```
   symstar -cg StarGrp reset -site London
   ```

The resulting SRDF/Star states are shown in Figure 76:

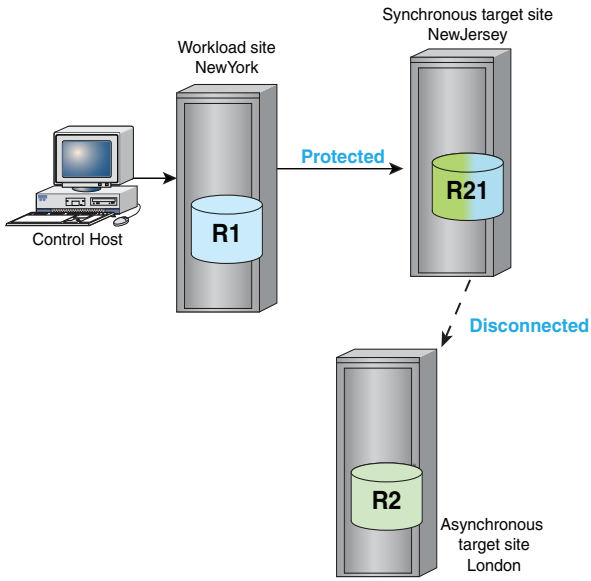


Figure 76 Cascaded SRDF/Star: asynchronous site not protected

Recovering from transient faults with reconfiguration: cascaded SRDF/Star

Note: Performing this operation changes the STAR mode of operation from cascaded to concurrent.

If:

- The asynchronous target site is no longer accessible, but
- The workload site is still operational, and
- The asynchronous target site is accessible through the recovery SRDF group,

You can:

- Reconfigure the SRDF/Star environment, and
- Resynchronize data between the workload site and the asynchronous target site to
- Achieve *direct* SRDF/A consistency protection between the workload site and the asynchronous target site.

Figure 75 on page 284 shows cascaded SRDF/Star with the workload site at `NewYork`, and a fault between the synchronous target site (NewJersey, and the asynchronous target site (London). The SRDF states are as follows:

- Synchronous target site (NewJersey): *Protected*
- Asynchronous target site (London): *PathFail*
- STAR state: *Tripped*

The first step varies depending on the state of the links to the asynchronous target site (London).

- If the links to the asynchronous target are in the *TransmitIdle* state, proceed to Step 1.

- If the links to the asynchronous target are in the *PathFail* state, skip to Step 2.

1. Use the `symstar -cg GroupName disconnect -trip` command to transition links to the asynchronous site to the *PathFail* state.

   `symstar -cg StarGrp disconnect -trip -site London`

2. Use the `symstar reconfigure` command from the workload site (NewYork) control host.

   See "symstar reconfigure command syntax" and "Usage notes" on page 268

   To reconfigure CG StarGrp as concurrent with the new SRDF pairs on the workload site (NewYork) and asynchronous target site (London), and perform a reset action:

`symstar -cg StarGrp reconfigure -reset -site London -path NewYork:London`

Note: If the system was not *STAR Protected*, you must use the `-full` option to perform full resynchronization.

The resulting SRDF/Star states are shown in Figure 77:

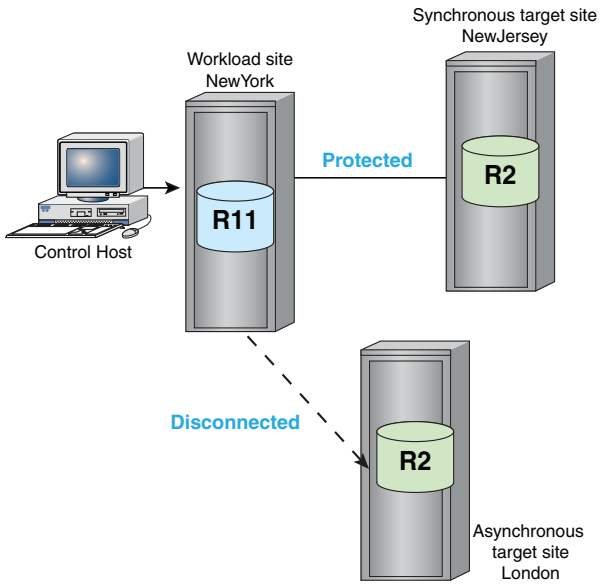


Figure 77 SRDF/Star: after reconfiguration to concurrent

Workload switching: cascaded SRDF/Star

This section describes the following topics:

- "Planned workload switching: cascaded SRDF/Star"
- "Unplanned workload switching: cascaded SRDF/Star"
- "Unplanned workload switch to synchronous target site: cascaded SRDF/Star"
- "Unplanned switch to the asynchronous target site: cascaded SRDF/Star"

Planned workload switching: cascaded SRDF/Star

Maintenance, testing and other activities may require you to switch the production workload site to another site.

This section describes the steps to switch workload sites when the operation can be scheduled in advance.

This operation requires you to:

- Stop the workload at the current production site,
- Halt the SRDF/Star environment (draining and synchronizing both remote sites in order for all three sites to have the same data), and
- Switching the production workload site to one of the remote sites.

If you switch the workload to the synchronous target site, you can transition to the STAR Protected state.

There is limited support for this configuration.

When configured as cascaded SRDF with the workload at London:

- Only the asynchronous link can be protected.
- The synchronous link (NewJersey -> NewYork) can only be connected.
- SRDF/Star cannot be enabled at London.

At the end of the switch operation the system comes up in the same STAR mode of operation that was configured before the `switch` operation was initiated.

1. At the current workload site (NewYork), use the SRDF/Star `halt` action.

   To halt the CG StarGrp:

   `symstar -cg StarGrp halt`

   The `halt` action:

   - Disables the R1 devices
   - Waits for all invalid tracks and cycles to drain,
   - Suspends the SRDF links,
   - Disables SRDF consistency protection, and
   - Sets the STAR mode indicators.

The target sites transition to the Halted state, with all three sites having the data, as shown in Figure 78:

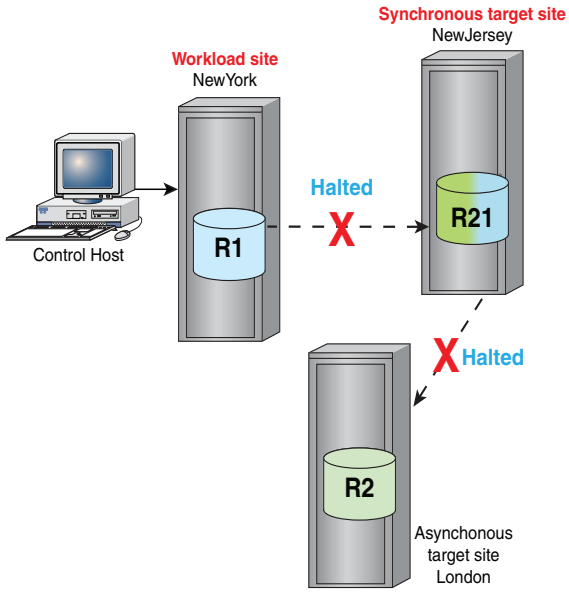


Figure 78 Cascaded SRDF/Star: halted

2. From a control host at the synchronous target site (`NewJersey`), use the `switch` command to switch the workload to the synchronous target site (`NewJersey`).

   ```
   symstar -cg StarGrp switch -site NewJersey
   ```

   The resulting SRDF/Star state is shown in Figure 79:

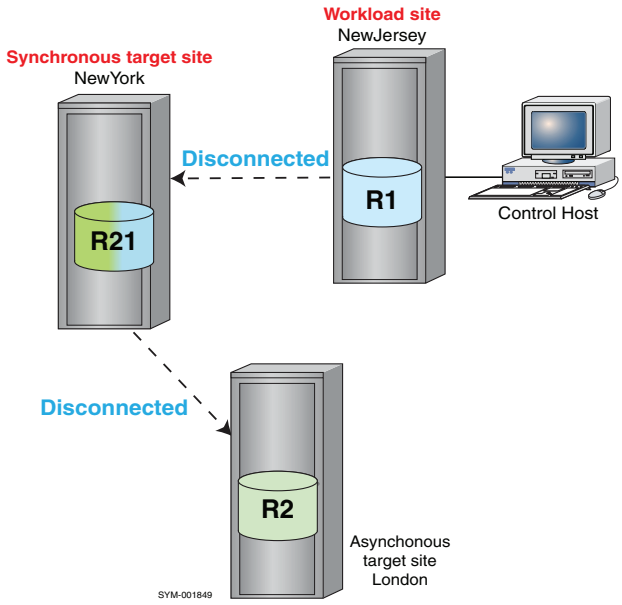


Figure 79 Cascaded SRDF/Star: switched workload site

Note: The entire SRDF/Star environment can also be halted from a non-workload site.

Unplanned workload switching: cascaded SRDF/Star

This section describes the procedure for switching the workload site to the synchronous site because of an unplanned event, such as a hurricane, causing the current workload site to stop processing I/Os.

This type of operation assumes the system is STAR Protected.

Note: There is limited support when switching from `NewYork` to `London`. When configured as cascaded SRDF/Star with the workload at London, only the long-distance link can be protected. The short-distance link can only be connected. SRDF/Star cannot be enabled at London.

Unplanned workload switch to synchronous target site: cascaded SRDF/Star

In cascaded mode, data at the synchronous target site is always more current than the data at asynchronous target site.

IMPORTANT

You cannot retain the data at the asynchronous target site if you move the workload to the synchronous target site.

In the following example, loss of the workload site (`NewYork`), has resulted in a system state of *NewJersey:Pathfail*:

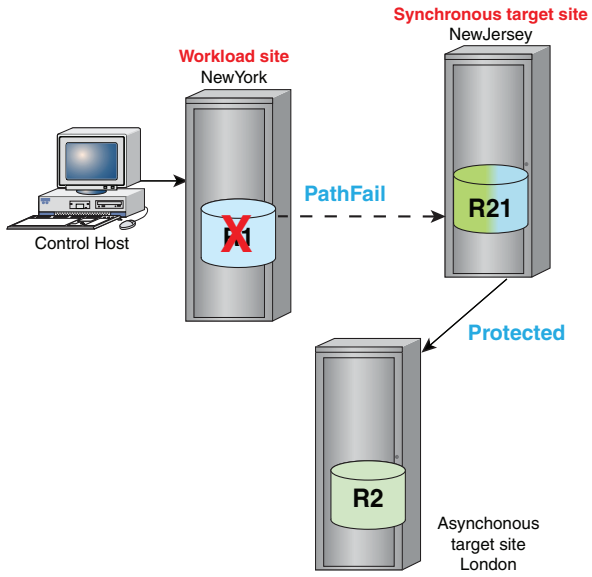


Figure 80 Loss of workload site: cascaded SRDF/Star

1. The first step varies depending on the state of the asynchronous target site (London).
   - If the asynchronous target site (London) is in Disconnected or PathFail state, skip to Step 2.
   - If the asynchronous target site (London) is in Protected state, issue a `disconnect` command from a control host at the synchronous target site (NewJersey) to get the asynchronous site to the *PathFail* state:

```
symstar -cg StarGrp disconnect -trip -site London
```

2. From a control host at the synchronous target site (`NewJersey`), use the `symstar cleanup` command to clean up any internal metadata or cache remaining at the asynchronous site.

   To clean up the London site:

   ```
   symstar -cg StarGrp cleanup -site London
   ```

3. From a control host at the synchronous target site (`NewJersey`), use the `symstar switch` command to start the workload at the specified site. The following command:

   - Specifies `NewJersey` as the new workload site (`-site NewJersey`)
   - Retains the data at the `NewJersey` data instead of the `London` data (`-keep_data NewJersey`):

   ```
   symstar -cg StarGrp switch -site NewJersey -keep_data NewJersey
   ```

   The resulting SRDF/Star state is shown in Figure 81:

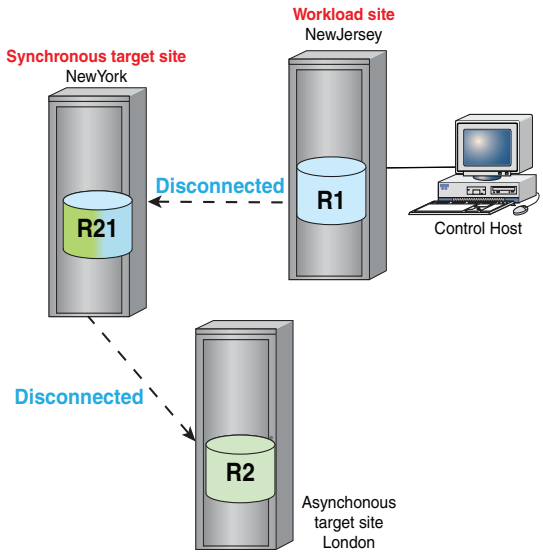


Figure 81 Workload switched to synchronous target site: cascaded SRDF/Star

4. If you are protecting your data with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

   For details, see "Step 7: (optional) Add BCV devices to the SRDF/Star configuration" on page 245.

5. After the switch, you can bring up SRDF/Star in a cascaded mode or reconfigure to come up in concurrent mode. The following examples explain the steps required for each mode:

   - Proceed to Step 6 to bring up SRDF/Star in cascaded mode (the default).
   - Skip to Step 8 to reconfigure SRDF/Star in concurrent mode.

6. From a control host at the new workload site (`NewJersey`), use two `connect` commands to:

   - Connect `NewJersey` to `NewYork` (synchronously)

- Connect `NewYork` to `London` (asynchronously):

```
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp connect -site London
```

The resulting SRDF/Star state is shown in Figure 82:

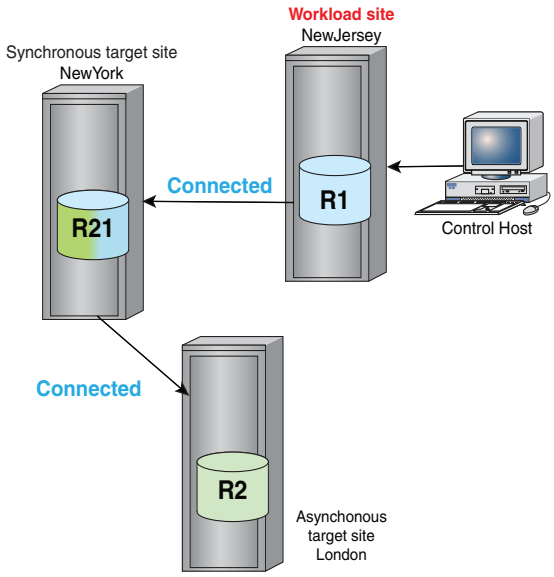


Figure 82 After workload switch to synchronous site: cascaded SRDF/Star

7. From a control host at the new workload site (`NewJersey`), use two `protect` commands and the `enable` command to:
   - Protect `NewJersey` to `NewYork`
   - Protect `NewJersey` to `London`
   - Enable SRDF/Star

```
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The resulting SRDF/Star state is shown in Figure 83:

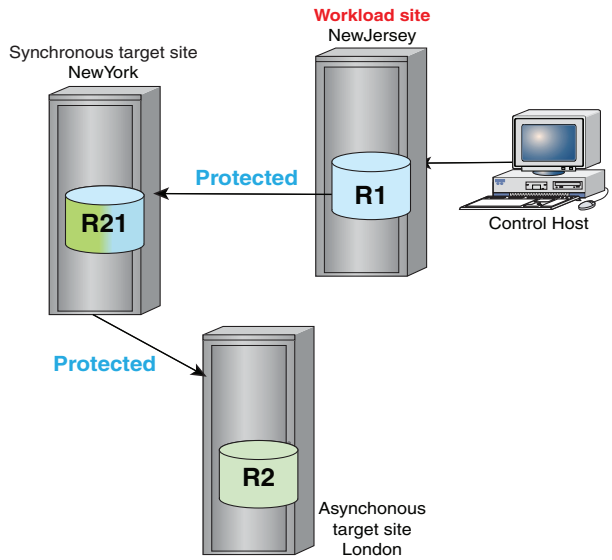


Figure 83 Cascaded SRDF/Star after workload switch: protected

8. From a control host at the new workload site, use the `symstar reconfigure` command from the workload site to change the mode to concurrent.

   See "symstar reconfigure command syntax" on page 267

   To reconfigure SRDF/Star to operate in concurrent mode with:

   - The workload at `NewJersey`,
   - The synchronous target site at `NewYork`, and
   - The asynchronous target site at `London`:

   ```
   symstar -cg StarGrp reconfigure -site London -path NewJersey:London
   ```

   The resulting SRDF/Star configuration is shown in Figure 84:

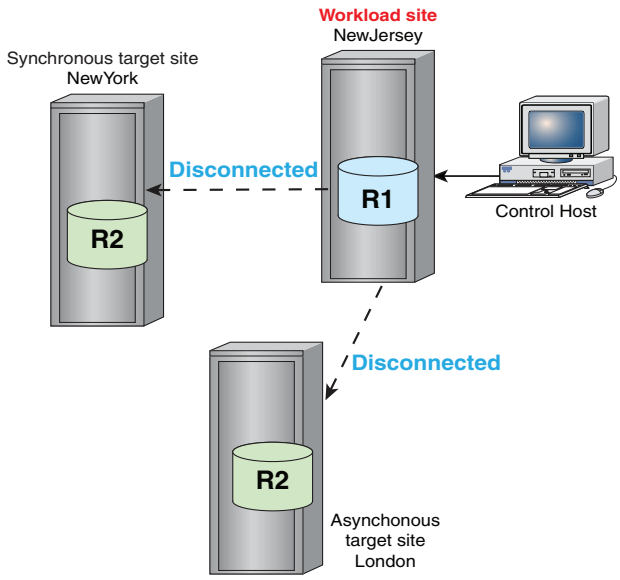


Figure 84 After reconfiguration to concurrent mode

9. Run the following commands from a control host at the new worload site (NewJersey) to:
   - Connect NewJersey to NewYork (synchronously)
   - Connect NewJersey to London (asynchronously)
   - Protect NewJersey to NewYork
   - Protect NewJersey to London
   - Enable SRDF/Star

```
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The resulting SRDF/Star configuration is shown in Figure 84:

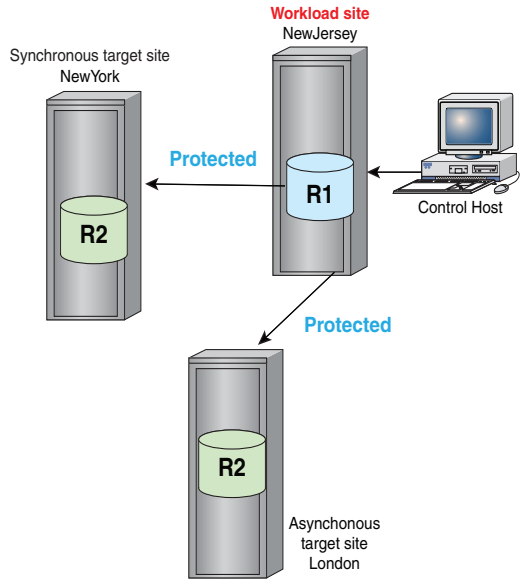


Figure 85 Protected after reconfiguration from cascaded to concurrent mode

Unplanned switch to the asynchronous target site: cascaded SRDF/Star

This section describes two procedures to switch the workload to the asynchronous target site:

- "Switch workload site: keep asynchronous site's data"
- "Switch workload site: keep synchronous site's data"

Switch workload site: keep asynchronous site's data

In the following example, the workload site (`NewYork`) has been lost:

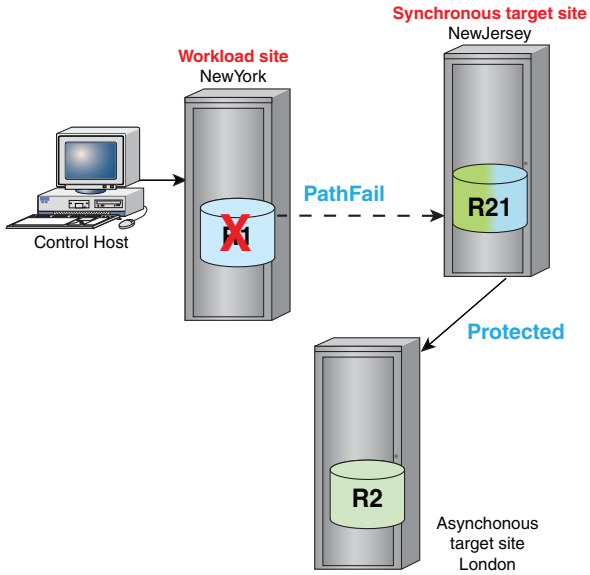


Figure 86 Loss of workload site: cascaded SRDF/Star

From a control host at the asynchronous target site (London), perform the following steps to:

- Switch the workload site to London
- Keep the data from the asynchronous target site (London):

1. If `London` is in a Protected state, use the `disconnect` command:

   ```
   symstar -cg StarGrp disconnect -trip -site London
   ```

2. If the `disconnect` leaves `London` in a CleanReq state, use the `cleanup` command:

   ```
   symstar -cg StarGrp cleanup -site London
   ```

3. Use the `switch` command to switch the workload site to the asynchronous target site (London) and keep the asynchronous target's (London's) data:

   ```
   symstar switch -cg StarGrp -site London -keep_data London
   ```

4. The London site is at asynchronous distance from both NewYork and NewJersey. SRDF/Star supports only one asynchronous site.

   When the `NewYork` site is repaired, you cannot connect *and* protect NewYork without switching the workload back to a configuration that has only one asynchronous site (`NewYork` or `NewJersey`).

   However, you can *connect* to NewYork. The `connect` action sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links.

   Use two `connect` commands to connect the workload site (London) to both target sites (NewJersey and New York):

   ```
   symstar -cg StarGrp connect -site NewJersey
   symstar -cg StarGrp connect -site NewYork
   ```

5. Use a `protect` command protect one target site (NewJersey):

   `symstar -cg StarGrp protect -site NewJersey`

   The resulting SRDF/Star configuration is shown in Figure 87:

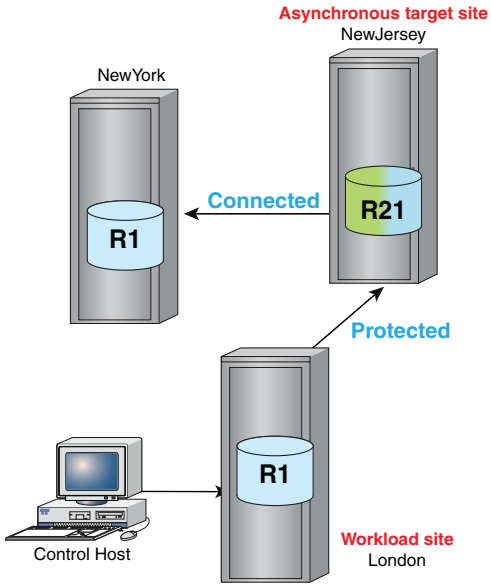


Figure 87 Cascaded SRDF: after switch to asynchronous site, connect, and protect

If you are protecting your data with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

"Step 7: (optional) Add BCV devices to the SRDF/Star configuration" on page 245.

"Switch back to the original workload site: concurrent SRDF/Star" on page 281 describes the steps to switch the workload site back to the initial site (NewYork).

Switch workload site: keep synchronous site's data

From a control host at the asynchronous target site (London), perform the following steps to:

- Switch the workload site to London
- Keep the data from the synchronous target site (NewJersey):

1. If `London` is in a Protected state, use the `disconnect` command:

   `symstar -cg StarGrp disconnect -trip -site London`

2. If the `disconnect` leaves `London` in a CleanReq state, use the `cleanup` command:

   `symstar -cg StarGrp cleanup -site London`

3. Use the `switch` command to switch the workload site to the asynchronous target site (London) and keep the synchronous target's (NewJersey) data:

   `symstar switch -cg StarGrp -site London -keep_data NewJersey`

   The workload site switches to London and the R2 devices at London become R1 devices.

   The London site connects to the NewJersey site and retrieves the NewJersey data.

Note: The `connect` action is not required because the `switch` action specified that SRDF retrieve the remote data from the NewJersey site.

The resulting SRDF/Star state is shown in Figure 71:

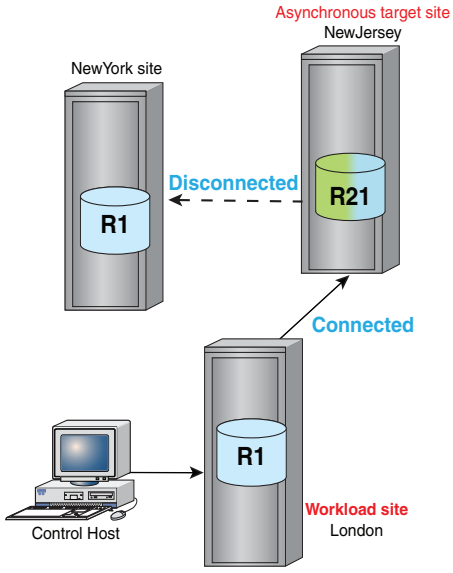


Figure 88 Cascaded SRDF: after switch to asynchronous site

If you are protecting your data with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

"Step 7: (optional) Add BCV devices to the SRDF/Star configuration" on page 245.

Reconfiguration operations

This section describes the following topics:

- "Reconfiguring mode: cascaded to concurrent"
- "Reconfiguring cascaded paths"
- "Reconfiguring mode: concurrent to cascaded"
- "Reconfiguring mode without halting the workload site"

Before you begin

- Reconfiguration of the STAR mode of operation is allowed only from the *Halted: Halted* state and leaves the system in *Halted: Halted* state.
- When the workload site is at NewYork or NewJersey, you can only reconfigure the path to the asynchronous target site.
- When the workload site is at London, you can reconfigure the path to either the synchronous target site or the asynchronous target site.
- If you do not want to halt the workload site, see "Reconfiguring mode without halting the workload site" on page 303.

Reconfiguring mode: cascaded to concurrent

This section describes the following topics:

- "Change mode to concurrent: from synchronous workload site"
- "Change mode to concurrent: from asynchronous workload site"

Change mode to concurrent: from synchronous workload site

1. From a control host at the workload site, use the halt command to stop SRDF:

   symstar -cg StarGrp halt

   The resulting SRDF/Star state is shown in Figure 89:

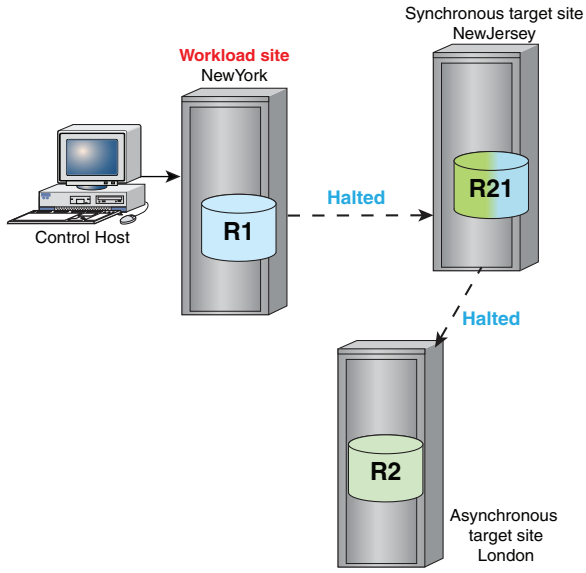


Figure 89 Halted cascaded SRDF/Star

2. Use the `symstar reconfigure` command to reconfigure the NewYork-› NewJersey -› London path to NewYork -› London:

   symstar -cg StarGrp reconfigure -site London -path NewYork:London

   See "symstar reconfigure command syntax" on page 267

The resulting SRDF/Star state is shown in Figure 90:

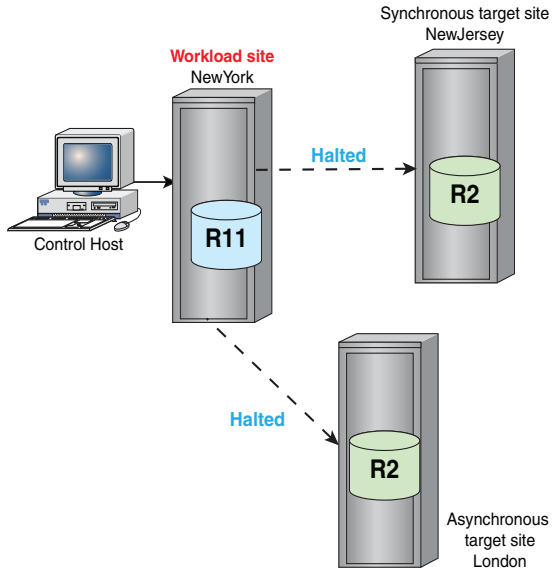


Figure 90 After reconfiguration to concurrent

Change mode to concurrent: from asynchronous workload site

1. From a control host at the workload site, use the halt command to stop SRDF:

   ```
   symstar -cg StarGrp halt
   ```

   The resulting SRDF/Star state is shown in Figure 91:

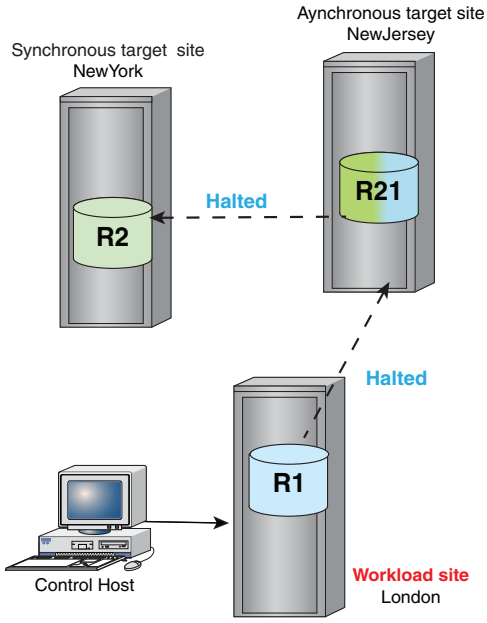


Figure 91 Halted cascaded SRDF/Star

2. Use the `symstar reconfigure` command to reconfigure the London› NewJersey -› NewYork path to London -› NewYork:

   ```
   symstar -cg StarGrp reconfigure -site NewYork -path London:NewYork
   ```

See "symstar reconfigure command syntax" on page 267

The resulting SRDF/Star state is shown in Figure 92:

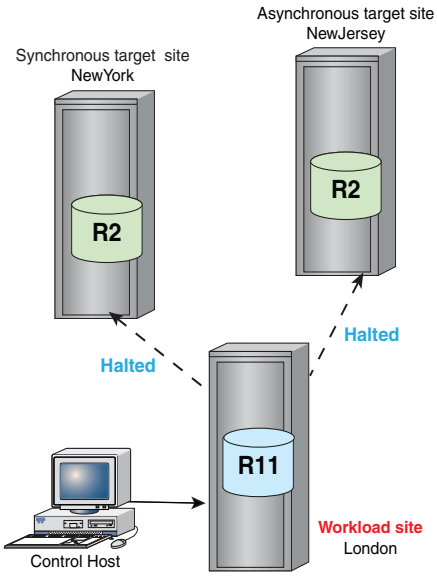


Figure 92 After reconfiguration to concurrent

Reconfiguring cascaded paths

In the following example:

- Both remote target sites are long distance sites from the workload site.
- The asynchronous target site is directly connected to the workload site.
- The other site is connected to the asynchronous target site is the synchronous target site.

Use the following steps to reconfigure the path to the synchronous target site (NewJersey) when the workload site is at `London`.

1. From a control host at the workload site, use the halt command to stop SRDF:

```
symstar -cg StarGrp halt
```

The resulting SRDF/Star state is shown in Figure 93:

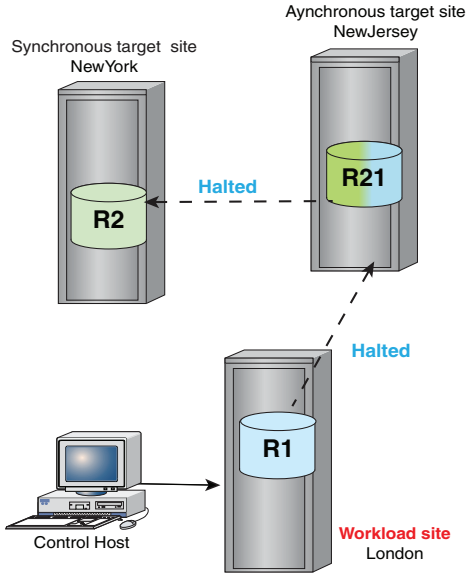


Figure 93 Halted cascaded SRDF/Star

2. Use the `symstar reconfigure` command with `-path` and `-remove` options to reconfigure the path from:

   London -> NewJersey -> NewYork

   to:

   London -> NewYork -> NewJersey:

   ```
   symstar -cg StarGrp reconfigure -site NewYork -path London:NewYork
   -remove London:NewJersey
   ```

   See "symstar reconfigure command syntax" on page 267

   The resulting SRDF/Star state is shown in Figure 94:

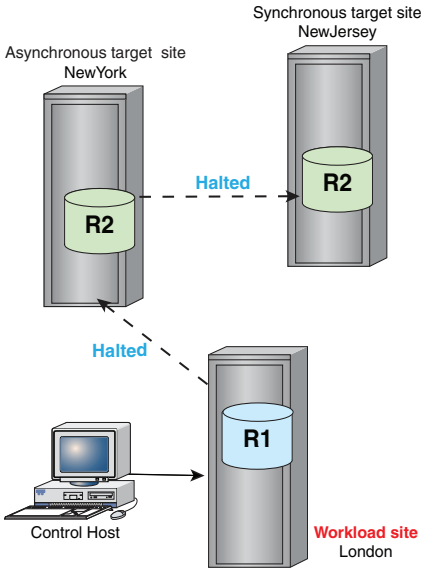


Figure 94 After cascaded path reconfiguration

Reconfiguring mode: concurrent to cascaded

This section describes the following topics:

- "Change mode to cascaded: from synchronous workload site"
- "Change mode to cascaded: from asynchronous workload site"

Change mode to cascaded: from synchronous workload site

1. From a control host at the workload site, use the halt command to stop SRDF:

   ```
   symstar -cg StarGrp halt
   ```

   The resulting SRDF/Star state is shown in Figure 95:

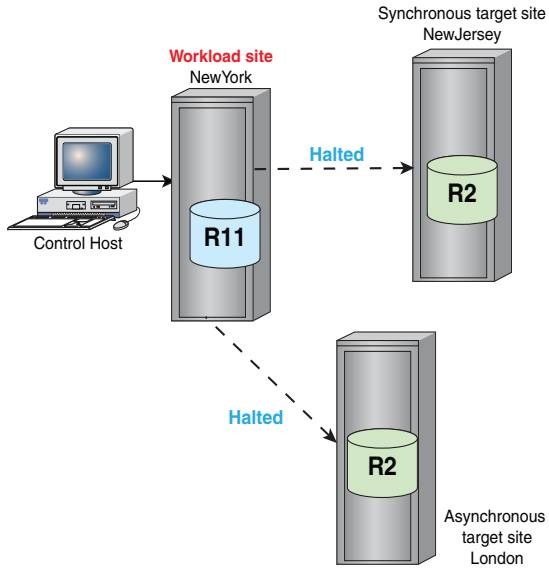


Figure 95 Halted concurrent SRDF/Star

2. Use the `symstar reconfigure` command to reconfigure the path from NewYork-> London to NewYork -> NewJersey -> London:

   ```
   symstar -cg StarGrp reconfigure -site London -path NewJersey:London
   ```

   See "symstar reconfigure command syntax" on page 267

The resulting SRDF/Star state is shown in Figure 96:

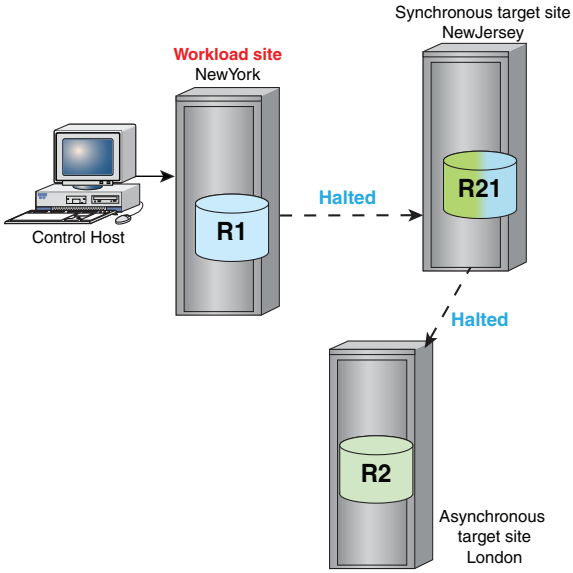


Figure 96 After reconfiguration to cascaded

Change mode to cascaded: from asynchronous workload site

1. From a control host at the workload site, use the halt command to stop SRDF:

   ```
   symstar -cg StarGrp halt
   ```

   The resulting SRDF/Star state is shown in Figure 91:

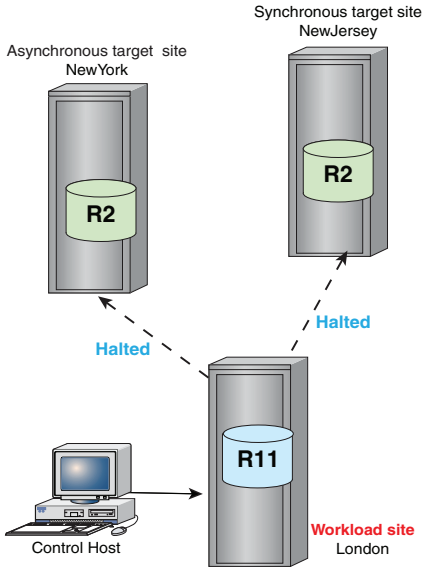


Figure 97 Halted concurrent SRDF/Star

2. Use the `symstar reconfigure` command to reconfigure the concurrent path from London -> NewYork to cascaded path London -> NewJersey -> NewYork:

   ```
   symstar -cg StarGrp reconfigure -site London -path NewJersey:London
   ```

   See "symstar reconfigure command syntax" on page 267

The resulting SRDF/Star state is shown in Figure 98:

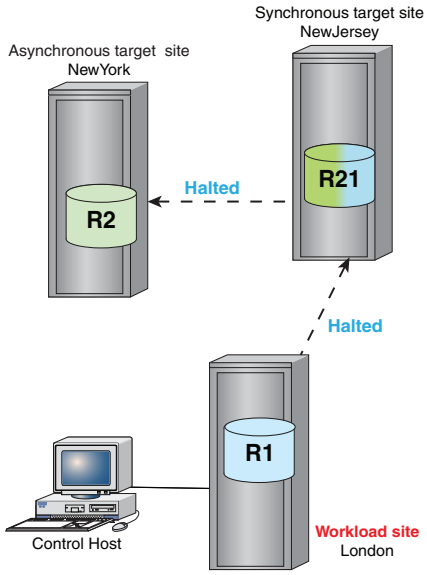


Figure 98 After reconfiguration to cascaded

Reconfiguring mode without halting the workload site

This section describes the following topics:

- "Reconfiguring cascaded mode to concurrent mode"
- "Reconfiguring concurrent mode to cascaded mode"

You can inject an disconnect/trip error to suspend the SRDF links to the asynchronous target site, and then follow the steps outlined in "Recovering from transient faults with reconfiguration: cascaded SRDF/Star" on page 285.

IMPORTANT

These operations take the system out of the "STAR Protected" state.

Once reconfiguration is complete, re-enable STAR protection.

Reconfiguring cascaded mode to concurrent mode

In the following example:

- The SRDF/Star environment is cascaded mode
- States are: Protected Sync, Protected Async, and Protected STAR
- The workload is at `NewYork`.
- The `symstar disconnect` command drops the links between `NewJersey` and `London`,
- The `reconfigure` changes the mode to concurrent:

```
symstar -cg StarGrp disconnect -trip -site London
symstar -cg StarGrp reconfigure -reset -site London -path NewYork:London
```

Note: Always follow `-trip` with `reconfigure -reset`.

Reconfiguring concurrent mode to cascaded mode

In the following example:

- The SRDF/Star environment is operating in concurrent mode
- States are: Protected Sync, Protected Async, and Protected Star
- The workload is at `NewYork`.
- The `symstar disconnect` drops the links between `NewYork` and `London`,
- The `reconfigure` changes the mode to cascaded.

```
symstar -cg StarGrp disconnect -trip -site London
symstar -cg StarGrp reconfigure -reset -site London -path NewJersey:London
```

SRDF/Star configuration with R22 devices

This section describes the following topics:

- "Before you begin"
- "Transitioning SRDF/Star to use R22 devices"

Before you begin

When creating an SRDF/Star configuration with R22 devices, verify/perform the following:

- The STAR compatibility mode must be set to v70 (the default value).

  `SYMCLI_STAR_COMPATIBILITY_MODE=v70`

  See "Step 4: Create the SRDF/Star options file" on page 242.

- All devices at the workload site must be configured as concurrent (R11) devices with one mirror paired with the R2 mirror of the remote R21 device (sync target site) and the other mirror paired with an R2 mirror of the remote R22 device (asynchronous target site).
- All devices at the sync target site must be configured as R21 devices paired with an R1 remote partner at the workload site and an R2 remote partner at the asynchronous target site.
- All devices at the asynchronous target site must be configured as R22 devices paired with an R21 remote partner at the sync target site and an R11 remote partner at the workload site.
- Create the appropriate RDF1 composite group (CG), adding the devices to the CG, setting RDFG names, etc. Note that in contrast to other SRDF/Star configurations, recovery SRDF groups do not need to be set in the CG for concurrent configurations.
- Once the configuration is ready, execute the `symstar setup` command using the `-opmode` option to choose either `concurrent` or `cascaded` operation.

The `symstar setup` command is allowed if the following SRDF pair states are Suspended, Synchronized, and SyncInProg:

- workload to sync target,
- workload to asynchronous target, or
- workload to sync target,
- sync target to asynchronous target site

Example

```
symstar -cg StarGrp setup -options MyOptnFile.txt -opmode concurrent

A STAR Setup operation is
in progress for composite group StarGrp. Please wait...

Setup ...............................................Started
Reading options file options.txt ....................Started
Reading options file options.txt ....................Done
Analyzing Host Composite Grp: r22cg .................Started
Syncing    Symmetrix information ....................Started
Syncing    Symmetrix information ....................Done
Gathering  Symmetrix SID: 000192600077 RDFG: 66......Started
Gathering  Symmetrix SID: 000192600077 RDFG: 66......Done
Gathering  Symmetrix SID: 000192600077 RDFG: 67......Started
Gathering  Symmetrix SID: 000192600077 RDFG: 67......Done
...
Distributing setup information to remote sites ......Started
Distributing setup information to remote sites ......Done
Update persistent state information .................Started
Update persistent state information .................Done
Setup ...............................................Done
```

Transitioning SRDF/Star to use R22 devices

You can transition an existing SRDF/Star environment to use R22 devices if the following are true:

- The current SRDF/Star environment is operating in normal condition.
- All sites must be reachable
- Relationships between the workload site and target sites must be properly configured.

Use the `symstar configure` command from the workload site:

```
symstar -cg CgName configure -add recovery_rdf_pairs
      [-opmode concurrent|cascaded]
```

This command is allowed from the workload site only while in the following states:

- Disconnected/Connected/Halted (to sync target site) and
- Disconnected/Connected/Halted (to asynchronous target site)

After the `configure` command completes, target sites are in the same states as they were in when the `configure` command was issued.

Example To immediately upgrade SRDF/Star to use R22 devices:

```
symstar -cg StarGrp configure -add recovery_rdf_pairs -opmode cascaded

A STAR Configure operation is
in progress for composite group StarGrp. Please wait...

Configure: Adding Recovery RDF Pairs................. Started
Update persistent state information ................. Started
Update persistent state information ................. Done
SA Write Disable Devs SID:000192600090............... Started
```

```
SA Write Disable Devs SID:000192600090............... Done
Createpair SID:000192600083 RDFG:114................. Started
Createpair SID:000192600083 RDFG:68.................. Started
Createpair SID:000192600083 RDFG:114................. Done
Createpair SID:000192600083 RDFG:68.................. Done
SA Write Enable Devs SID:000192600090................ Started
SA Write Enable Devs SID:000192600090................ Done
Distributing setup information to remote sites .......Started
Distributing setup information to remote sites .......Done
Update persistent state information ................. Started
Update persistent state information ................. Done
Configure: Adding Recovery RDF Pairs ................ Done
```

Use the `symstar show` command to verify R22 devices are configured as the recovery SRDF pairs. For example (truncated output):

```
Composite Group Name    : StarGrp

   Recovery RDF Pairs Configured   : Yes

    Site SiteA to site SiteB Information:
```

Use the `symstar query` command to verify that adding recovery RDF pairs was the last action performed. For example (truncated output):

symstar -cg *CgName* query

```
...
Last Action Performed           :ConfigureAddRcvryRDFPair
Last Action Status              :Successfull
Last Action timestamp           :03/15/2008_12:29:37
```

CHAPTER 9
Device Migration Operations

This chapter describes the following topics:

- Overview 308
- Requirements 308
- R1 device migration 309
- R2 device migration 312
- Restrictions and limitations 316
- Sample procedure: migrating R1 devices 317
- Sample procedure: migrating R2 devices 325
- Pair states for migrate -setup 328
- Pair states for migrate -replace for first leg of concurrent SRDF 329
- Pair states for migrate -replace for second leg of concurrent SRDF 331

Overview

SRDF device migration allows you to replace an existing device in an SRDF pair with a new device on a different array.

During migration, a concurrent SRDF relationship is established to transfer data from an existing R1 device to a new device in adaptive copy disk mode.

When data transfer completes, the R1 device or the R2 device is replaced with the newly-populated device in the SRDF pair.

Requirements

- Each array must have a unique ID (sid).
- The existing SRDF device and the new devices must be dynamic R1 or R2 capable.

HYPERMAX OS

- Adaptive copy write pending mode is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.

  For configurations where the R1 side is on an array running HYPERMAX OS, and the R2 side is running Enginuity 5876, the mode of the new device pair is set to the RDF mode of the R1 device being replaced.

- The Geometry Compatibility Mode attribute (-gcm) allows devices on arrays running HYPERMAX OS to be paired with devices on arrays running Enginuity 5876 that have an odd number of cylinders. When GCM is set, migration operations are subject to the following restrictions:
  - If the new device is on an array running HYPERMAX OS:
    - If the R1 device is being replaced:

      If the existing R2 device is on an array running Enginuity 5876 with an odd number of cylinders, then the migration is allowed if the new device can be made the same size using the GCM attribute.

      If the existing R2 device is on an array running HYPERMAX OS with GCM set, then the migration is allowed if the new device can be made the same size by setting the GCM attribute.

    - If the R2 is being replaced:

      If the existing R1 device is on an array running Enginuity 5876 with an odd number of cylinders, then the migration is allowed if the new device can be made the same size by setting the GCM attribute.

      If the existing R1 device is on an array running HYPERMAX OS with GCM set, then the migration is allowed if the new device can be made the same size by setting the GCM attribute.
  - If the new device is on an array running Enginuity 5876 and has an odd number of cylinders:
    - If the R1 is being replaced:

      If the existing R2 device is on an array running Enginuity 5876, then the new device must be the same configured size.

If the existing R2 device is on an array running HYPERMAX OS with GCM set, then the migration is allowed if the new device has the same GCM size as the R2 device.

– If the R2 is being replaced:

  If the existing R1 device is on an array running Enginuity 5876, then the new device must be the same configured size.

  If the existing R1 device is on an array running HYPERMAX OS with GCM set, then the migration will be allowed if the new device has the same GCM size as the R1.

R1 device migration

Before you can migrate an R1 device to a new array, you must create a temporary concurrent SRDF configuration with the new array as one of the R2 sites.

This section describes the steps to complete an R1 migration, including:

- "Configure a temporary SRDF group" and R1 device to enable the migration.
- "Establish a concurrent SRDF relationship"to transfer data to the from the old R1 device to the device that will become the new R1.
- "Replace the R1 device" with the newly-populated device in the SRDF pair.

Configure a temporary SRDF group

Configure a temporary SRDF group to synchronize data from the existing R1 device to the new R1 device.

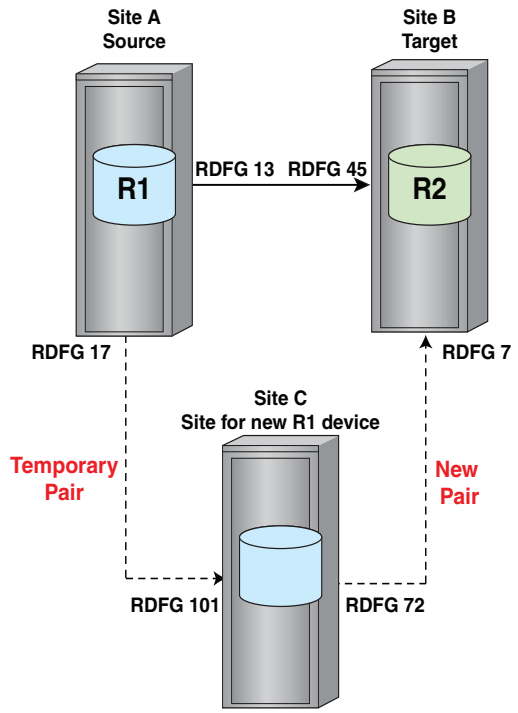


Figure 99 R1 migration: configuration setup

In Figure 99:

- Site A contains the existing R1 device paired with the R2 device in Site B,
- Site C contains the new non-SRDF device you want replace the existing R1 device.

  The dotted lines indicate that there are no SRDF relationships to Site C.
- A temporary SRDF group (RDFG 17) is used to synchronize data from the existing R1 to the new device in Site C.

The new R1 device replaces the existing R1 device during the migration.

Establish a concurrent SRDF relationship

Use the `symrdf migrate -setup` command to establish a concurrent relationship between the source device and two target devices.

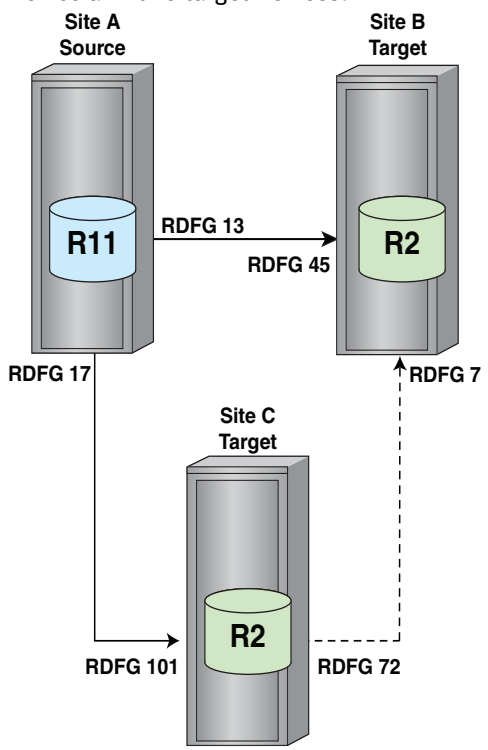


Figure 100 R1 migration: establishing a concurrent relationship

In Figure 100:

- The R1 device becomes the concurrent R11 device writing to two R2devices.
- Data synchronization in adaptive copy disk mode begins between the device and the R2 device on Site C.
- No SRDF pairing exists between the devices on Site C and Site B.

Note: You may need to modify existing device group or composite group scripts to accommodate the new R11 configuration.

Replace the R1 device

1. Wait until the two R2 devices are near synchronization with the R11 device.
2. Shut down any applications writing to the source device.

3. Use the `symrdf migrate -replace R1` command to replace the source device.

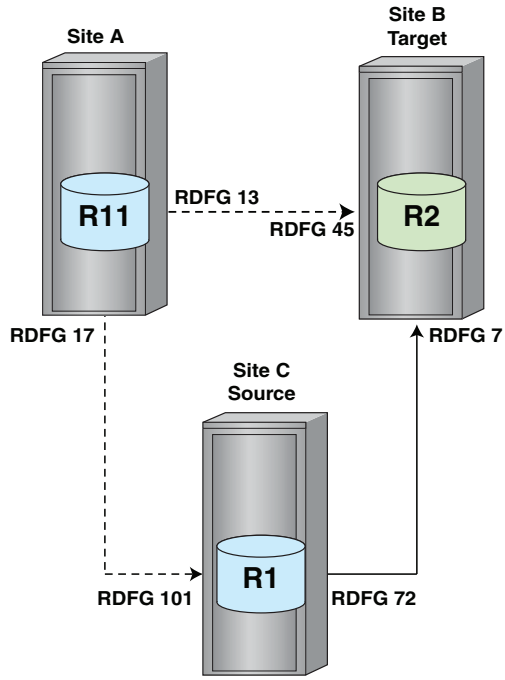


Figure 101 R1 migration: replacing the source device

The `symrdf migrate -replace R1` command executes the following actions:

1. Sets the source device to USR-NR (user not ready).

   This prevents applications writing to or reading from the R1 device.

2. Verifies the devices are in the correct pair state for replacement.

   See also "SRDF pair states for migration" on page 327.

3. (If applicable) Waits until all invalid tracks are cleared.
4. (If applicable) Drains the SRDF/A session.
5. Removes the SRDF pairing between the devices on the current R11 (Site A) and the original R2 (Site B).
6. Removes the SRDF pairing between the devices on the current R11 (Site A) and the new R2 (Site C).
7. Sets an SRDF pairing between the devices on Site C and B using the original SRDF mode of Site A and B.

   No additional copying of data is required between this SRDF pair because data is already the same on both devices.

8. Makes the devices read/write on the SRDF links.

The new R1 device is ready. You can restart the applications writing to the new R1 device on Site C.

The original R1 device remains USR-NR.

R2 device migration

R2 device migration allows you to replace the original R2 devices with new R2 devices. It shows the initial two-site topology, the migration process, and the final SRDF topology.

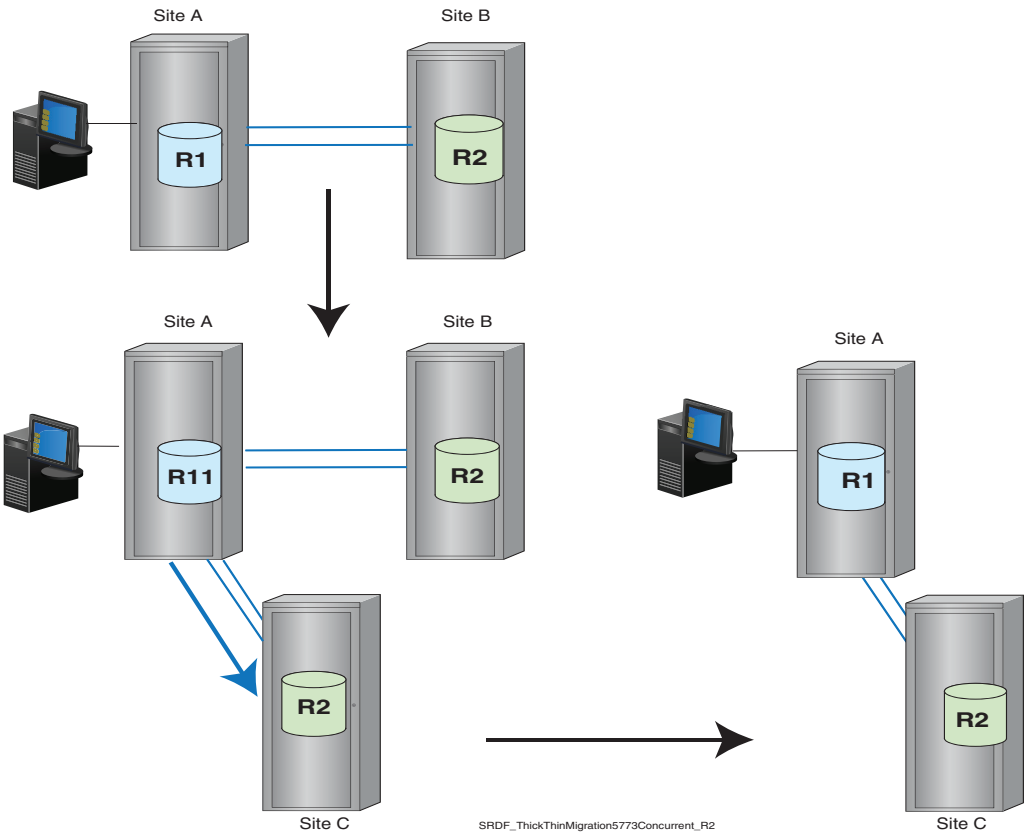


Figure 102 Migrating R2 devices

This section describes the steps to complete an R2 migration, including:

- "Configure a replacement R2 as a non-SRDF device:"
- "Establish a concurrent SRDF relationship"to transfer data to the from the R1 device to the device that will become the new R2.
- "Replace the R2 device" with the newly-populated device in the SRDF pair.

Configure setup for R2 migration

Configure a replacement R2 as a non-SRDF device:

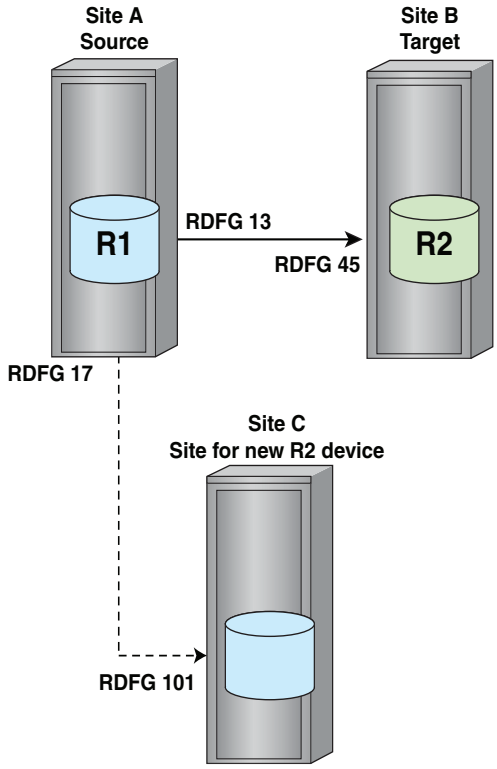


Figure 103 R2 migration: configuration setup

In Figure 103:

- Site A contains the R1 device paired with the existing R2 device in Site B,
- Site C contains the new non-SRDF device that will replace the R2 device.

  The dotted lines indicate no SRDF pairing exists with Site C.

Establish a concurrent SRDF relationship

Use the `symrdf migrate -setup` command to establish a concurrent SRDF relationship among the three sites:

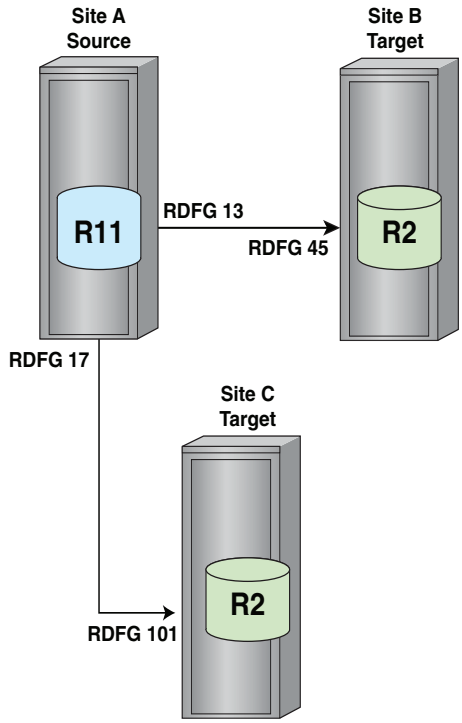


Figure 104 R2 migration: establishing a concurrent relationship

The establish action creates a concurrent SRDF relationship to transfer data from the existing source device to both target devices.

In Figure 104, the R1 becomes the R11 device writing to two target R2 devices.

- The source site continues to accept I/Os from the host.
- There is no need to shut down the applications writing to R1.
- No temporary pairing (like an R1 migration) is required.
- The source and target devices do not have to be close to synchronization.

Note: You may need to modify existing device group or composite group scripts to accommodate the new configuration.

Replace the R2 device

Use the `symrdf migrate -replace R2` command to replace the existing R2 device with the new R2 device in the SRDF pair:

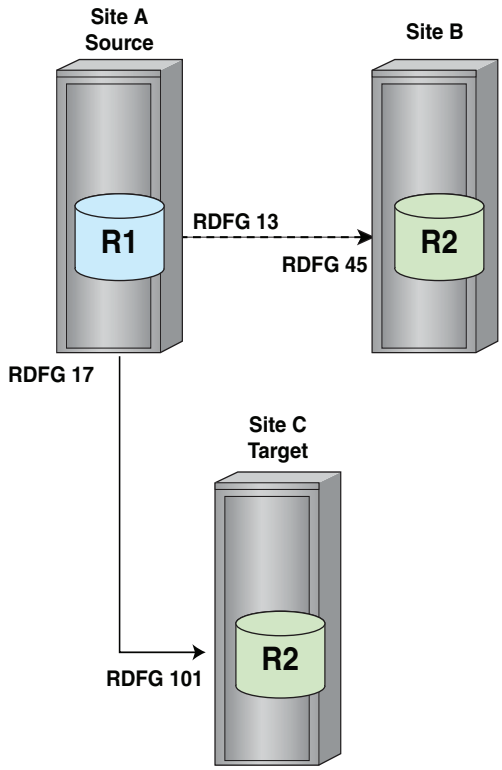


Figure 105 R2 migration: replacing the target device

The `symrdf migrate -replace R2` command executes the following actions:

1. Verifies the devices are in the correct pair state for replacement.

   "SRDF pair states for migration" on page 327 provides more information.

2. Removes the SRDF pairing between the devices on Site A and B.
3. Sets the mode of Site A and C using the original SRDF mode of Site A and B.

R1 and R2 migration procedures

Before you begin

- Plan for each migration.

  If you have defined scripts for your existing R1/R2 pair, evaluate how you may need to modify those scripts with new SIDs, SRDF device pairings, device groups, and composite groups.

  Keep in mind that during a device migration, the R1/R2 pair transforms into a concurrent SRDF relationship (R2<-R11->R2), and then back into an R1->R2 relationship.

- An SRDF group must exist for the new device.

If R1 is being replaced, this is the SRDF group between the new R1 and the existing R2.

If R2 is being replaced, this is the SRDF group between the new R2 and the existing R1.

- For an R1 migration *only*, a temporary SRDF group is required to synchronize data from the existing R1 device to the new device.

  If performing an R1 migration, create this temporary SRDF group.

- Before replacing the R1 device, you must shut down all applications using it.

  Application shutdown is not required when replacing an R2 device.

- Review "SRDF pair states for migration" on page 327.

Restrictions and limitations

SRDF/A device pairs

- The attributes associated with an existing SRDF group pertaining to an SRDF/A session are not automatically associated with the new SRDF group after migration.

  You must issue the `symconfigure` command on the new SRDF group and set the appropriate attributes, such as the minimum_cycle_time and the DSE (Delta Set Extension) autostart settings.

- If replacing a device of an SRDF pair in SRDF/A mode, all existing rules for DSE apply if DSE autostart is enabled on the new SRDF group.

  For example, the DSE threshold must be less than the maximum cache usage for the new SRDF group.

- If replacing the R1 device of an SRDF pair in SRDF/A mode, the new SRDF group in the new R1 array must be SRDF/A capable.
- If replacing a device of an SRDF pair in SRDF/A mode and Cache partitioning is enabled on the new array, all new devices must belong to the same cache partition.
- If the existing device is in SRDF/A mode, the entire SRDF group must be migrated.
- If the existing device is in SRDF/A mode, the new SRDF group must be empty.
- If replacing the R1 device, the temporary SRDF group must not be in SRDF/A mode.
- The existing SRDF device pair cannot be in semi-synchronous mode.

Devices

- The new device (R1 or R2) cannot be an SRDF device before migration.
- The existing device (R1 or R2) and the replacement device cannot be diskless.
- The new R1 device cannot be larger than the existing R1 device.
- The existing R1 device cannot have any local invalid tracks.
- After migration, the R2 device cannot be larger than the R1 device.
- The existing (R1 or R2) and the new device cannot be configured for SRDF/Star.
- The existing device and the replacement device cannot be a source or a target device for TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration.

This restriction does not apply to the SRDF partner of the existing device.

- The existing R1/R2 device pair cannot be in a concurrent SRDF relationship.

  Set the `-config` option to equal `pair` in `symrdf migrate -setup` to indicate this pair is not part of such a configuration.

- An SRDF consistency protection group must be enabled at the RDFG-name level, **NOT** at the composite-group level.

  Otherwise, the `migrate -setup` command stops the monitoring/cycle switching of your composite group.

  "Sample procedure: migrating R1 devices", explains the procedure for an SRDF consistency protection group enabled at the composite-group level.

Sample procedure: migrating R1 devices

For this sample procedure, the SRDF consistency protection group is enabled at the composite-group level.

This procedure shows the steps to change this setting and enable SRDF consistency protection at the RDFG-name level.

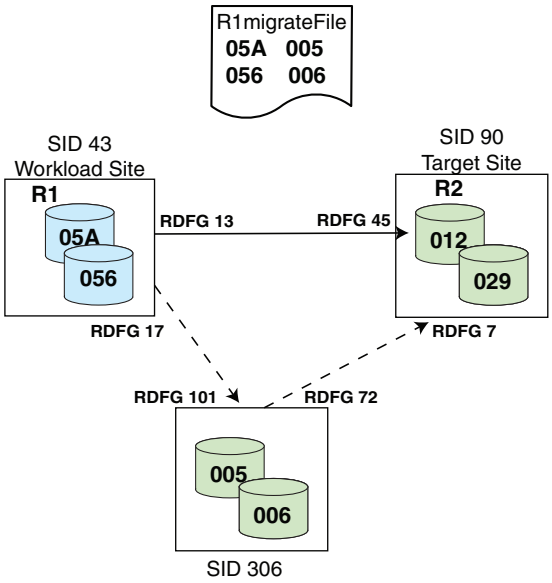


Figure 106 R1 migration example: Initial configuration

Figure 106 shows an R1 and R2 relationship between array 43 and array 90.

After R1 migration, the devices in array 306 will become the source devices for array 90.

Step 1: Querying the sample SRDF/A configuration

Use the `symrdf query -detail` command to query a configuration with SRDF consistency protection enabled at the composite-group level.

```
symrdf -cg MigrateRDF query -detail

Composite Group Name       :  MigrateRDF
Composite Group Type       :  RDF1
Number of Symmetrix Units  :  1
```

```
Number of RDF (RA) Groups :  1
RDF Consistency Mode      :  MSC

RDFA MSC Consistency Info:{
    Session Status                      : Active
    Consistency State                   : CONSISTENT
    }

Symmetrix ID                      : 000192600043 (Microcode Version: 5874)
Remote Symmetrix ID               : 000192600090 (Microcode Version: 5874)
RDF (RA) Group Number             :   1 (00) 13 (0C)
RDFA Info:
    {
    Cycle Number                        : 29
    Session Status                      : Active - MSC
    Consistency Exempt Devices          : No
    Minimum Cycle Time                  : 00:00:30
    Avg Cycle Time                      : 00:00:30
    Duration of Last cycle              : 00:00:30
    Session Priority                    : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1           : 00:00:42
    R2 Image Capture Time               : Mon Sep 21 13:28:44 2009
    R2 Data is Consistent               : True
    R1 Side Percent Cache In Use        : 0
    R2 Side Percent Cache In Use        : 0
    R1 Side DSE Used Tracks             : 0
    R2 Side DSE Used Tracks             : 0
    Transmit Idle Time                  : 00:00:00
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ------------
               ST                 LI       ST
  Standard      A                  N        A
Logical  Sym    T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev    E  Tracks Tracks   S Dev    E  Tracks  Tracks MDACE STATE
-------------------------------- -- ------------------------ ----- ------------
DEV001   0005A NR       0       0 RW 00012 WD       0       0 A..X. Consistent
DEV002   000F8 NR       0       0 RW 00029 WD       0       0 A..X. Consistent

Total            ------- -------            ------- -------
  Track(s)             0       0                  0       0
  MBs                0.0     0.0                0.0     0.0
```

Step 2: Changing the SRDF consistency protection setting

To maintain consistency protection after establishing a concurrent SRDF relationship:

- Remove the SRDF consistency protection enabled at the composite-group level, and then
- Enable consistency protection at the RDFG-name level.

In the following example:

- The `symcg set -name siteb` command sets the SRDF group name to `siteb`.
- The `symcg disable` command disables SRDF consistency protection at the composite-group level
- The `symcg enable` command enables SRDF consistency protection at the RDFG-name level.

```
symcg -cg MigrateRDF -rdfg 043:13 set -name siteb
symcg -cg MigrateRDF disable
```

```
A consistency 'Disable' operation execution is
in progress for composite group 'MigrateRDF'. Please wait...

The consistency 'Disable' operation successfully executed for
composite group 'MigrateRDF'.
```

```
symcg -cg MigrateRDF -rdfg name:siteb enable
```

```
A consistency 'Enable' operation execution is
in progress for composite group 'MigrateRDF'. Please wait...

The consistency 'Enable' operation successfully executed for
composite group 'MigrateRDF'.
```

Verifying the changes

Use the `symrdf query -detail` command to verify that the changes and additions were made to the SRDF/A configuration.

In the following example, SRDF consistency protection is now enabled using the SRDF group name of `siteb`.

```
symrdf -cg MigrateRDF query -detail

Composite Group Name      :  MigrateRDF
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  1
RDF Consistency Mode      :  NONE

RDFG Names:
    {
    RDFG Name                          : siteb
    RDF Consistency Mode               : MSC
    MSC Consistency Info:
        {
        Session Status                 : Active
        Consistency State              : Consistent
        }
    }
```

Step 3: Pairing devices

Create a device file to pair SRDF devices with the new non-SRDF devices.

"Creating a device file" on page 94 provides more information.

This pairing is used temporarily to transfer data from the existing R1 devices to the devices that will eventually replace them in an SRDF pair.

In the following example, device file R1MigrateFile contains two pairs:

```
05A 005
056 006
```

R1 devices 05A and 056 in array 43 are paired with the new devices 005 and 006 in array 306.

Step 4: Establishing a concurrent SRDF relationship

The `symrdf migrate -setup` command establishes a concurrent SRDF relationship between the existing R1 devices and the new devices in adaptive copy disk mode, and begins the synchronization of these devices.

Note: You may need to modify existing device group or composite group scripts to accommodate the temporary change of the existing R1 devices to R11 devices.

The `symrdf -migrate -setup -config pair -force` command establishes a concurrent SRDF relationship between the R1 devices in array 43 and the new devices in array 306 using SRDF group 17.

This is a temporary relationship to transfer data from the existing R1 to its replacement.

Using the -force option

You must use the `-force` option because SRDF consistency protection is enabled.

```
symrdf -sid 043 -rdfg 17 -f R1MigrateFile migrate -setup -config pair -force

An RDF 'Migrate Setup' operation execution is
in progress for device file 'R1migrateFile'. Please wait...

    Migrate Setup for R1 device(s) in (043,017).......................Started.
    Create RDF Pair in (0043,017)....................................Started.
    Create RDF Pair in (0043,017)....................................Done.
    Mark target device(s) in (0043,017) for full copy from source....Started.
    Devices: 06F0-06FF in (0043,017)............................... Marked.
    Mark target device(s) in (0043, 017) for full copy from source...Done.
    Merge track tables between source and target in (0043,017).......Started.
    Devices: 06F0-06FF in (0043,017)............................... Merged.
    Merge track tables between source and target in (0043,017).. ....Done.
    Resume RDF link(s) for device(s) in (0043,017)...................Started.
    Resume RDF link(s) for device(s) in (0043,017)...................Done.
    Migrate Setup for R1 device(s) in (0043,017) ....................Done.

The RDF 'Migrate Setup' operation finished successfully
for device file 'R1MigrateFile'.
```

Note: If the host is reading and writing to the R1 device during this action, a synchronized pair state may not be attainable because the pair is operating in adaptive copy disk mode.

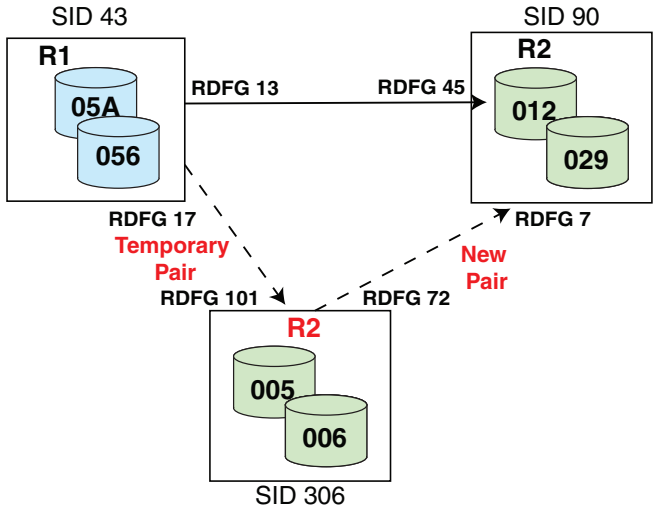


Figure 107 Concurrent SRDF relationship

In Figure 107:

- Devices 05A and 056 are paired with devices 005 and 006 in a concurrent SRDF relationship using SRDF group 17.
- Devices 005 and 006 are made read/write on the SRDF links in adaptive copy disk mode.
- SRDF group 17 is used temporarily to transfer data from the R1 devices to the new devices.

Step 5: Replacing R1 devices with new devices

1. If consistency is enabled, use the `symcg disable` command to disable it.

   To disable SRDF consistency protection for composite group `MigrateRDF`:

   ```
   symcg -cg MigrateRDF -rdfg name:siteb disable

   A consistency 'Disable' operation execution is
   in progress for composite group 'MigrateRDF'. Please wait...

   The consistency 'Disable' operation successfully executed for
   composite group 'MigrateRDF'.
   ```

2. Terminate any TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration sessions.
3. Use the `symrdf migrate -replace` command to set R1 (R11) device as USR-NR, complete the final synchronization of data between the existing and the new device, and reconfigure the devices into a new SRDF pair.

   The device pairings of the replaced devices are removed. The new devices become R1 devices paired with the existing R2 devices using the original SRDF mode of the replaced pair.

   Note: The `migrate -replace R1` command waits for synchronization to finish and may take a long time. To avoid the locking of the SYMAPI database for this entire time, set the environment variable SYMCLI_CTL_ACCESS=PARALLEL. If you set this variable, you may need to run the `symcfg sync` command after the R1 migration is complete.

In the following example, the `migrate -replace R1` command specifies the new SRDF group 72 to reconfigure and connect the new R1 devices 005 and 006 in array 306 with the R2 devices 012 and 029 in Symmetix 90:

```
symrdf -sid 043 -rdfg 17 -f R1migrateFile migrate -replace r1 -config pair -new_rdfg 72

An RDF 'Migrate Replace R1' operation execution is
in progress for device file 'R1migrateFile'. Please wait...

    Migrate Replace R1 for new R1 device(s) in (0306, 072)...........Started.
    Waiting for invalid tracks to reach 0 in   (0043, 013)...........Started.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........Started.
    Waiting for invalid tracks to reach 0 in   (0043, 013)...........Done.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........915994 remaining.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........519572 remaining.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........245889 remaining.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........107613 remaining.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........1110 remaining.
    Waiting for invalid tracks to reach 0 in   (0043, 017)...........Done.
    Suspend RDF link(s) for device(s) in (0043,013)..................Started.
    Suspend RDF link(s) for device(s) in (0041,013)..................Done.
    Suspend RDF link(s) for device(s) in (0043,017)..................Done.
```

```
    Delete RDF Pair in (0043,013)..................................Started.
    Delete RDF Pair in (0043,017)..................................Started.
    Delete RDF Pair in (0043,013)..................................Done.
    Delete RDF Pair in (0043,017)..................................Done.
    Create RDF Pair in (0306,072)..................................Started.
    Create RDF Pair in (0306,072)..................................Done.
    Resume RDF link(s) for device(s) in (0306,072)..................Started.
    Merge track tables between source and target in (0306,072).......Started.
    Devices: 0690-069F in (0306,072)............................... Merged.
    Merge track tables between source and target in (0306,072).......Done.
    Resume RDF link(s) for device(s) in (0306,072)..................Done.
    Migrate Replace R1 for new R1 device(s) in (0306, 072)..........Done.

The RDF 'Migrate Replace R1' operation finished successfully
for device file 'R1migrateFile'.
```

After replacing the R1 devices:

- Recreate your device groups and/or composite groups,
- Possibly update your scripts, since the devices are no longer concurrent SRDF.
- Recreate any TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration sessions (used on the original R1 devices) on the new R1 devices.

In the following example, the `MigrateRDF` consistency group is deleted and re-created:

- The `symcg delete` command deletes the `MigrateRDF` consistency group.
- The `symcg create` command recreates `MigrateRDF` as an RDF1 with consistency.
- The `symcg addall dev` command add devices `MigrateRDF`.
- The `symcg enable` command enables consistency protection.

```
symcg -force delete MigrateRDF
symcg create MigrateRDF -type rdf1 -rdf_consistency
symcg -cg MigrateRDF -sid 306 -rdfg 72 addall dev
symcg -cg MigratRDF enable

A consistency 'Enable' operation execution is
in progress for composite group 'MigrateRDF'. Please wait...

The consistency 'Enable' operation successfully executed for
composite group 'MigrateRDF'.
```

When migration is complete (as shown in Figure 108):

- SID 306 devices are the R1 devices.
- SID 306 devices are paired with the R2 devices in SID 90.

- This new SRDF pair uses the original SRDF mode of the replaced pair.

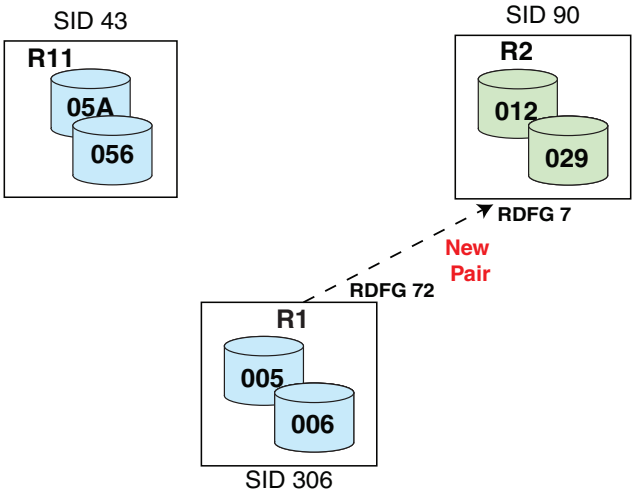


Figure 108 Migrated R1 devices

Step 6: Verifying the new pair and setting changes

Use the `symrdf query -detail` to verify that:

- The SID 306 devices are now the source devices for SID 90,
- Consistency protection is rebuilt.

```
symrdf -cg MigrateRDF query -detail

Composite Group Name       :  MigrateRDF
Composite Group Type       :  RDF1
Number of Symmetrix Units  :  1
Number of RDF (RA) Groups  :  1
RDF Consistency Mode       :  MSC

RDFG MSC Consistency Info:{
   Session Status                  : Active
   Consistency State               : CONSISTENT
    }

Symmetrix ID                       : 000190100306 (Microcode Version: 5773)
Remote Symmetrix ID                : 000192600090 (Microcode Version: 5874)
RDF (RA) Group Number              :   3 (02) - siteb
RDFA Info:
    {
    Cycle Number                        : 3
    Session Status                      : Active - MSC
    Consistency Exempt Devices          : No
    Minimum Cycle Time                  : 00:00:30
    Avg Cycle Time                      : 00:00:33
    Duration of Last cycle              : 00:00:30
    Session Priority                    : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1           : 00:00:34
    R2 Image Capture Time               : Mon Sep 21 13:52:03 2009
    R2 Data is Consistent               : True
    R1 Side Percent Cache In Use        : 0
    R2 Side Percent Cache In Use        : 0
    R1 Side DSE Used Tracks             : 0
    R2 Side DSE Used Tracks             : 0
    Transmit Idle Time                  : 00:00:00
```

```
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ------------
               ST                 LI       ST
  Standard      A                  N        A
Logical  Sym    T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev    E  Tracks Tracks   S Dev    E  Tracks  Tracks MDACE STATE
-------------------------------- -- ------------------------- ----- ------------
DEV001   00005 RW       0       0 RW 00012 WD       0       0 A..X. Consistent
DEV002   00006 RW       0       0 RW 00029 WD       0       0 A..X. Consistent

Total            ------- -------            ------- -------
  Track(s)             0       0                  0       0
  MBs                0.0     0.0                0.0     0.0
```

Sample procedure: migrating R2 devices

In this migration example, the devices in array 306 will become the R2 devices for array 43.

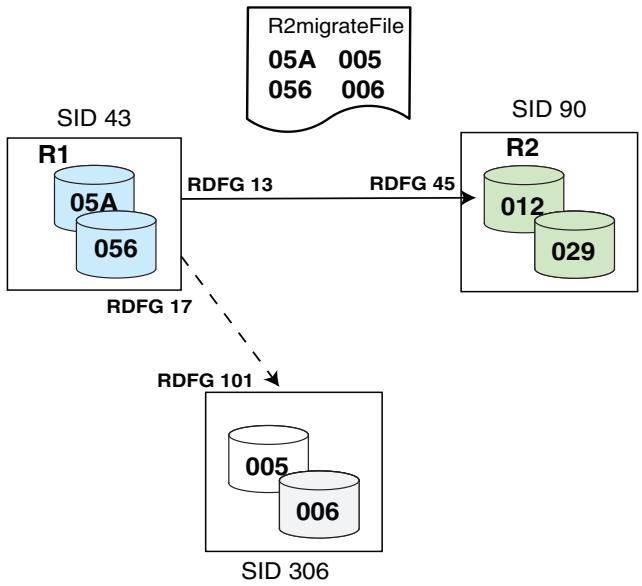


Figure 109 R2 migration example: Initial configuration

Figure 109 shows the R1 and R2 relationship between array 43 and array 90.

Step 1: Pairing devices

Create a device file to pair SRDF devices with the new non-SRDF devices.

"Creating a device file" on page 94 provides more information.

In the following example, device file R2MigrateFile contains two pairs:

```
05A 005
056 006
```

When migration is complete, R1 devices 05A and 056 in array 43 will be paired with the new devices 005 and 006 on array 306.

Step 2: Establishing a concurrent SRDF relationship

The `symrdf migrate -setup` command establishes a concurrent SRDF relationship between the existing R1 devices and the new devices in adaptive copy disk mode, and begins the synchronization of these devices.

Because this is an R2 migration, the R1 continues to process I/Os from its host, and synchronization is not required between the R1 and the new device.

Note: You may need to modify existing device group or composite group scripts to accommodate the temporary change of the existing R1 devices to R11 devices.

The `symrdf migrate -setup -config pair` command establishes a concurrent SRDF relationship between the R1 devices 05A and 056 in array 43 and the new devices 005 and 006 in array 306 using SRDF group 17:

```
symrdf -file R2migrateFile -sid 043 -rdfg 17 migrate -setup -config pair
```

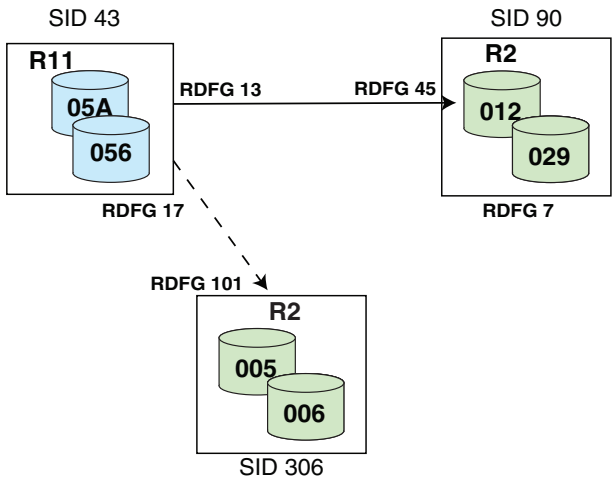


Figure 110 Concurrent SRDF relationship

In Figure 110:

- Devices 05A and 056 are paired with devices 005 and 006 in a concurrent SRDF relationship using the SRDF group 17,
- Devices 005 and 006 are made read/write on the SRDF links in adaptive copy disk mode.

Unlike an R1 device migration, the SRDF group 17 is permanent, and synchronizes data from the source to the target devices.

Step 3: Replacing R2 devices with new devices

1. If SRDF consistency protection is enabled, disable it.
2. Terminate any TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration sessions.
3. Use the `symrdf migrate -replace R2` command to delete the SRDF pairing between array 43 and array 90.

   Note: After replacing R2, you must modify device groups and/or composite groups to remove all BCVs, VDEVS, TGTs from the original R2 and then add appropriate counterparts to the new R2.

   You must also recreate any TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration sessions on the new R2.

In the following. example, the `symrdf migrate -replace R2 -config pair` command uses the SRDF group 17 to reconfigure and connect the R1 devices 05A and 056 with the new R2 devices 005 and 006:

```
symrdf -file R2migrateFile -sid 043 -rdfg 17 migrate -replace R2 -config pair.
```

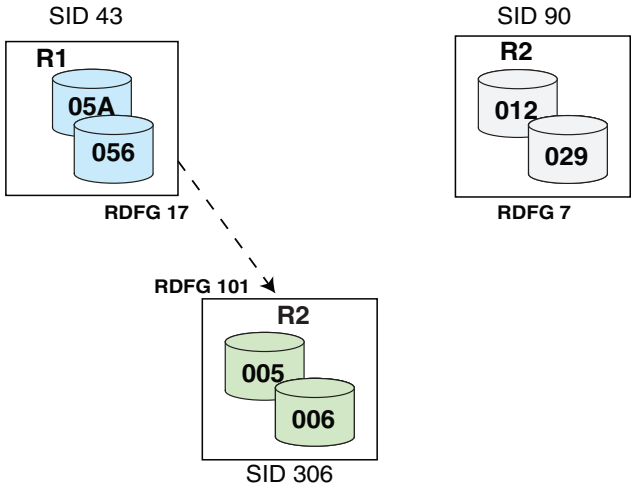


Figure 111 Migrated R2 devices

When migration is complete, the array 306 devices become the R2 devices and are paired with the R1 devices in Symmetix 43.

This new pair uses the original SRDF mode of the replaced pair.

SRDF pair states for migration

An existing R1 and R2 pair must in a specific SRDF state to perform certain migration control operations.

Table 37 shows the applicable pair states for `symrdf migrate -setup` for an R1 and an R2 migration.

Table 37 SRDF migrate -setup control operation and applicable pair states

Control operation:	Pair state: existing R1->R2											
	SYNC IN PROG	SYNCHRONIZED	SPLIT	SUSPENDED	FAILED OVER	PARTITIONED[1]	PARTITIONED[2]	R1 UPDATED	R1 UPD IN PROG	INVALID	CONSISTENT	TRANSMIT IDLE
migrate -setup	P	P	P[3]	P[3]							P	

1. The remote array is in the SYMAPI database (it was discovered).
2. The remote array is not in the SYMAPI database (it was not discovered or was removed).
3. Only when replacing the R2 devices.

Pair states for migrate -setup

Figure 112 shows a sample configuration for an R1 migration:

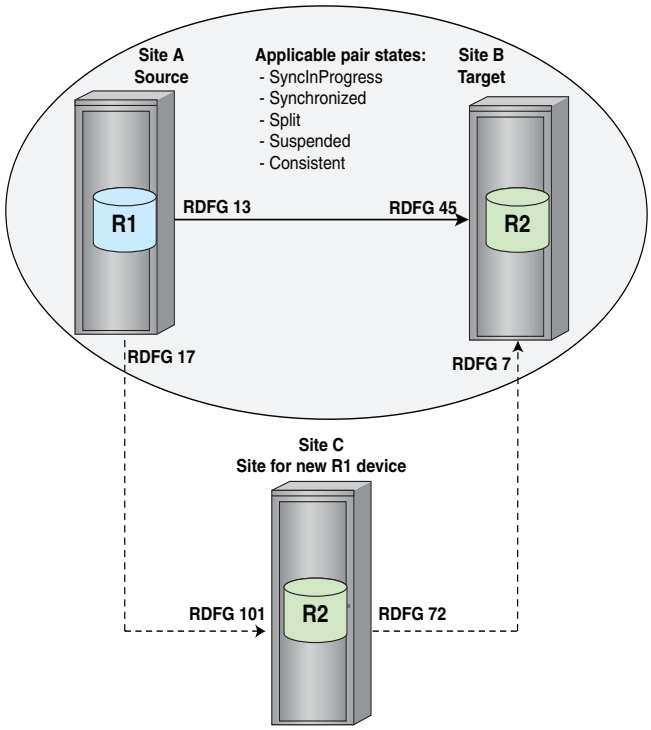


Figure 112 R1 migration: applicable R1/R2 pair states for migrate -setup

The R1 in array A and the R2 in array B must be in one of the applicable pair states before issuing the `symrdf migrate -setup` command, which establishes a concurrent SRDF relationship among the three sites.

Figure 113 shows a sample configuration for an R2 migration:

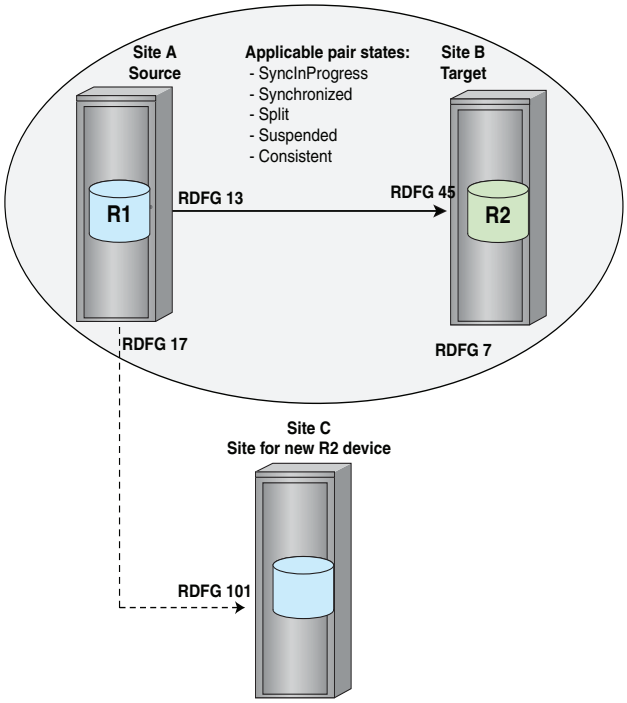


Figure 113 R2 migration: applicable R1/R2 pair states for migrate -setup

The R1 in array A and the R2 in array B must be in one of the applicable pair states before issuing the `symrdf migrate -setup` command, which establishes a concurrent SRDF relationship among the three sites.

Pair states for migrate -replace for first leg of concurrent SRDF

Figure 114 shows the SRDF pair state required before replacing an R1, the R11 and its existing device.

Figure 115 shows the SRDF pair state required when replacing R2, the R11 and its existing R2 device. For the purpose of this discussion, this is the first leg of the concurrent SRDF relationship for both R1 and R2 migrations.

Table 38 shows the applicable pair states for `symrdf migrate -replace` for an R1 and an R2 migration.

Table 38 SRDF migrate -replace control operation and applicable pair states

	Pair state: Existing ->R2											
Control operation:	SYNC IN PROG	SYNCHRONIZED	SPLIT	SUSPENDED	FAILED OVER	PARTITIONED[1]	PARTITIONED[2]	R1 UPDATED	R1 UPD IN PROG	INVALID	CONSISTENT	TRANSMIT IDLE
migrate -replace	P	P	P	P							P	

1. The remote array is in the SYMAPI database (it was discovered).

2. The remote array is not in the SYMAPI database (it was not discovered or was removed).

Figure 114 shows a sample concurrent SRDF configuration for an R1 migration:.

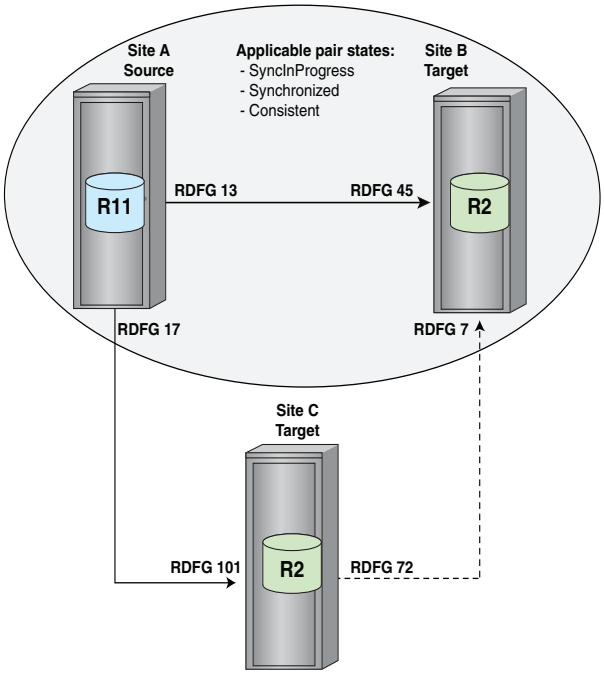


Figure 114 R1 migration: R11/R2 applicable pair states for migrate -replace (first leg)

The R11 in array A and the R2 device in array B must be in one of the applicable pair states before issuing the `symrdf migrate -replace` command.

Figure 115 shows a sample concurrent SRDF configuration for an R2 migration:

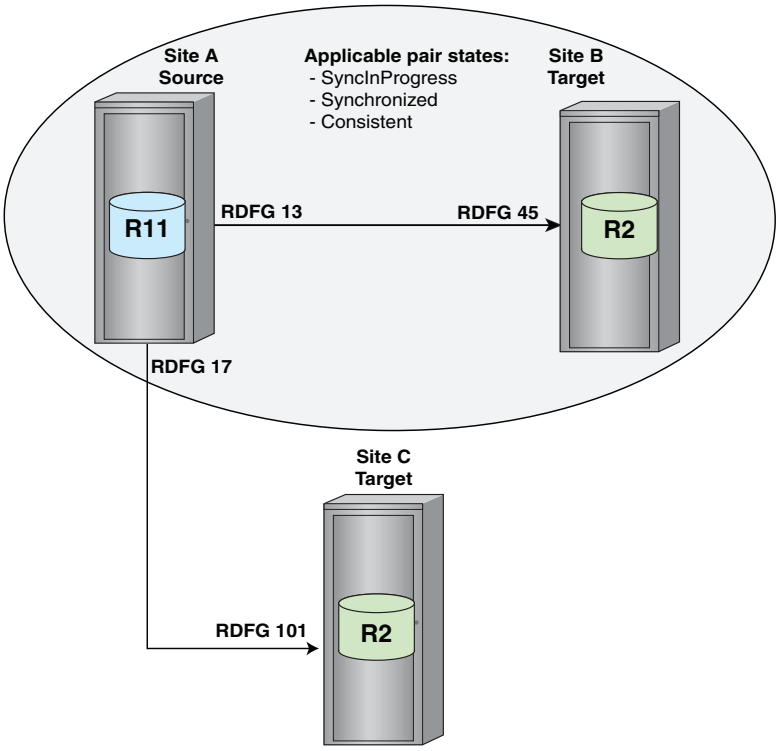


Figure 115 R2 migration:R11/R2 applicable pair states for migrate -replace (first leg)

The R11 in array A and the R2 device in array B must be in one of the states before issuing the `symrdf migrate -replace` command

Pair states for migrate -replace for second leg of concurrent SRDF

Before replacing an R1, the R11 and its replacement device must in a specific SRDF pair state shown in Figure 116. This temporary pairing was used to perform the concurrent SRDF data transfer to the new device. When replacing R2, the R11 and the new R2 device (new pair) must also be in a certain pair state shown in Figure 117.

Table 39 shows the applicable pair states for `symrdf migrate -replace` for an R1 and an R2 migration.

Table 39 SRDF migrate -replace control operation and applicable pair states

Control operation:	Pair state: Temporary or New ->R2											
	SYNC IN PROG	SYNCHRONIZED	SPLIT	SUSPENDED	FAILED OVER	PARTITIONED[1]	PARTITIONED[2]	R1 UPDATED	R1 UPD IN PROG	INVALID	CONSISTENT	TRANSMIT IDLE
migrate -replace	P	P									P	

1. The remote array is in the SYMAPI database (it was discovered).
2. The remote array is not in the SYMAPI database (it was not discovered or was removed).

Figure 116 shows a sample concurrent SRDF configuration for an R1 migration.

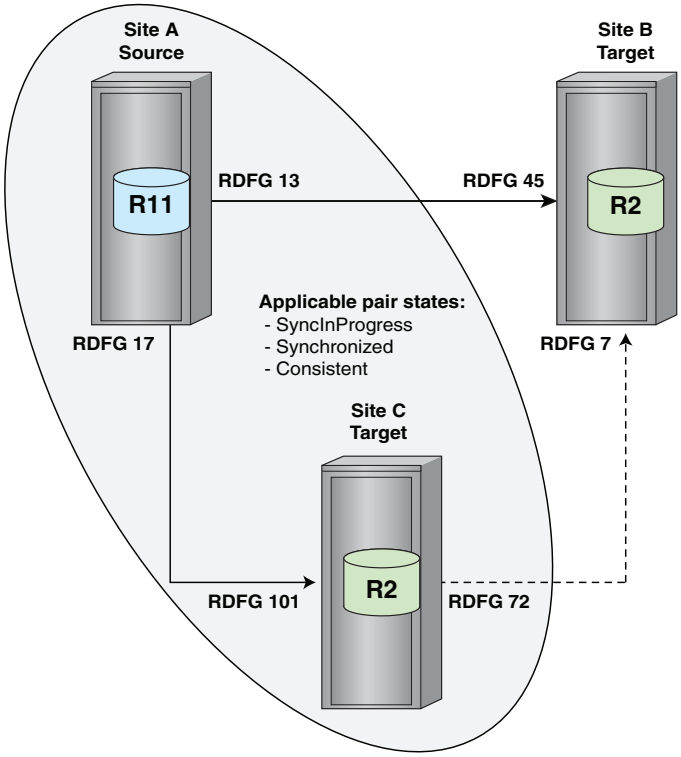


Figure 116 R1 migration: applicable R11/R2 pair states for migrate -replace (second leg)

The R11 device in array A and the R2 device in array C must be in one of the applicable pair states before issuing the `symrdf migrate -replace` command.

Figure 117 shows a sample concurrent SRDF configuration for an R2 migration:

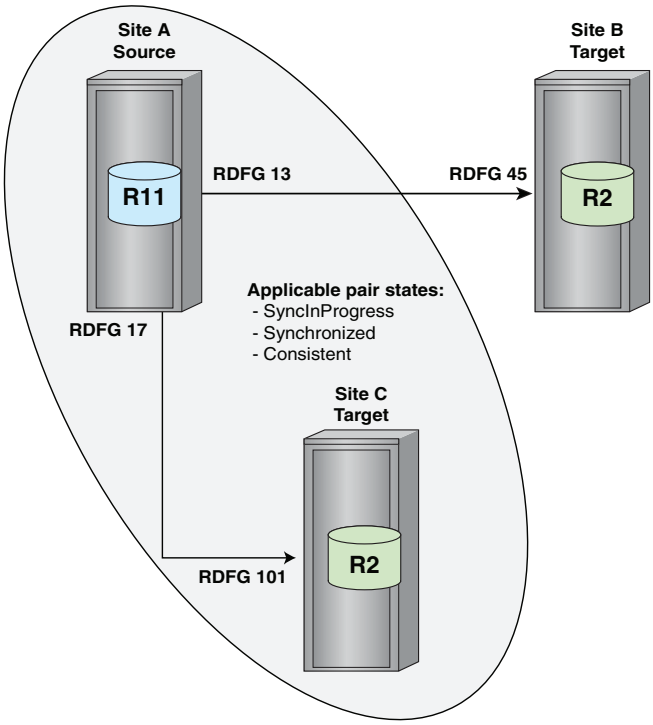


Figure 117 R2 migration: applicable R11/R2 pair states for migrate -replace (second leg)

The R11 in array A and the R2 device in array C must be in one of the states before issuing the `symrdf migrate -replace` command.

CHAPTER 10
SRDF/Automated Replication

This chapter describes the following topics:

- Overview 336
- Restrictions 336
- Configuring single-hop sessions 337
- Configuring multi-hop sessions 341
- Concurrent BCVs with SRDF/AR 343
- Setting replication cycle parameters 344
- Clustered SRDF/AR 346
- Setting symreplicate parameters in the options file 348
- Managing locked devices 354

Overview

SRDF/Automated Replication (SRDF/AR) provides a long-distance disaster restart solution. SRDF/AR can operate:

- In two-site topologies that use SRDF/DM in combination with TimeFinder.
- In three-site topologies that use a combination of SRDF/S, SRDF/DM, and TimeFinder.

  Three-site topologies operate in synchronous mode in the first hop and in adaptive copy mode in the second hop.

  Note: Note: Multi-hop SRDF/AR requires Enginuity version 5876.159.102 or higher.

SRDF/AR provides automated consistent replication of data from standard devices and RDF1 BCV devices over SRDF links to remote SRDF pairs.

SRDF/AR is invoked using the `symreplicate` command.

- `symreplicate` supports single-hop and multi-hop SRDF configurations.
- You can start, stop, or restart a `symreplicate` session without degrading the data copy.
- You can set up a concurrent BCV to have access to an independent copy of the replicating data during a `symreplicate` session.

By default, the `symreplicate` replication process is performed in the background.

Restrictions

- SRDF/AR does not support SRDF/Asynchronous-capable devices.
- The `symreplicate` command operates on device groups and composite groups.

  Scope for the `symreplicate` command cannot be limited to a specific SRDF group using the `-rdfg` option.
- When running `symreplicate` against device groups and composite groups of type ANY:
  - Concurrent SRDF devices are not supported for device groups (DG) or composite groups (CG).
  - The following combinations of standard devices are supported when using the `-consistent` option:
    - All STDs are non-SRDF
    - All STDs are R1 devices
    - All STDs are R2 devices
    - STDs contain a mixture of R1s and non-SRDF devices
    - STDs contain a mixture of R2 and non-SRDF devices

Note: Device external locks in the array are held during the entire `symreplicate` session. Locks are necessary to block other applications from altering device states while the session executes. "Managing locked devices" on page 354 provides more information.

SRDF/Automated Replication operations

Configuring single-hop sessions

Figure 118 shows how `symreplicate` copies data in a single-hop configuration for a complete copy cycle:

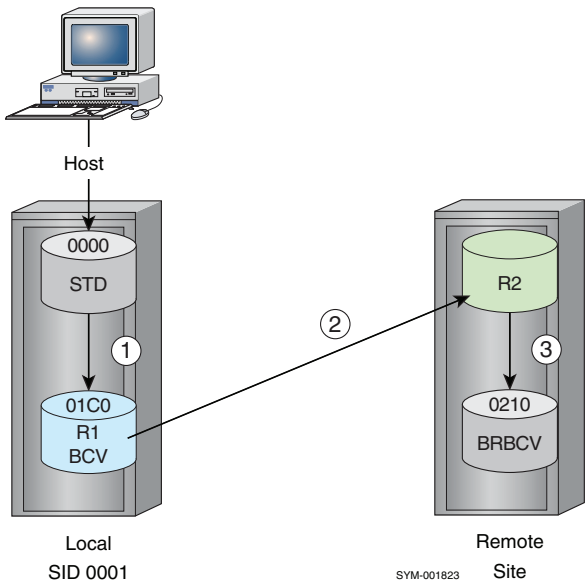


Figure 118 Automated data copy path in single-hop SRDF systems

The copy process includes the following steps:

- **1** - From the standard device to the BCV of the local array.
- **2** - From the BCV device of the local array to the standard device of the remote array.
- **3** - From the remote standard device to its BRBCV device.

Before you begin: setting the hop type parameter

You must set the replication type parameter in the replicate options file before you can configure a single-hop `symreplicate` session.

"SYMCLI_REPLICATE_HOP_TYPE=‹RepType›" on page 350provides more information.

Set the parameter as follows:

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
```

The `symreplicate` session:

- Incrementally establishes SRDF and BCV pairs, and

- Differentially splits BCV pairs to reduce data transfers.

Setting up single-hop data replication

To set up a single-hop `symreplicate` session:

1. Select any number of standard devices of the same type (R1, R2, or non-SRDF).
2. Use the `symdg create` command to create a device group or composite group of the same type.

   ```
   symdg create newdg
   ```

3. Use the `symdg add dev` command to add the devices to the device group.

   ```
   symdg add dev 0000 -g newdg -sid 35002
   symdg add dev 0001 -g newdg
   ```

4. Use the `symbcv associate` command to associate an equal number of R1-BCV devices of matching sizes.

   ```
   symbcv associate dev 01C0 -g newdg
   symbcv associate dev 01C1 -g newdg
   ```

5. Use the `symbcv associate` command to associate an equal number of BRBCV devices (remote BCVs), also of matching sizes.

   ```
   symbcv associate dev 0210 -g newdg -bcv -rdf
   symbcv associate dev 0211 -g newdg -bcv -rdf
       .
       .
       .
   ```

Note: The `symreplicate` command uses composite groups (`-cg`) to implement single-hop or multi-hop configurations for devices that span multiple arrays.

The following must be true before you start a `symreplicate` session:

- Both sets of BCV pairs must have a pairing relationship.
- The local BCV pairs must be established.
- The SRDF pairs must be in the Suspended pair state.
- The remote BCVs (BRBCVs) must be in the split pair state.
- No writes are allowed to the BRBCV by any directly attached host at the remote site.

Setting up pair states automatically

You can set up the required pair state pair for SRDF/AR automatically using either:

- `symreplicate setup` command
- `symreplicate start` command with the `-setup` option

Auto-replication setup sets up the required pair states for devices and executes one copy (auto-replication) cycle.

Setting up the device states ahead of time reduces replication processing time.

The `setup` commands execute one cycle of the `symreplicate` session (regardless of the number of cycles defined in the options file), and then exits.

The default setup operation provides no I/O optimization, and does not engage any special algorithm changes in the selection of pair assignments. For standard devices encountered without BCVs, the first unassigned BCV device found is paired with the standard.

Setup operations correct only pair states of devices in the group. If a BCV in the group is paired with a standard device outside of the group, setup does not correct it.

The setup command does not exit until the devices are in the required pair state to run the `symreplicate` session. This may take some time.

Note: Optionally, you can manually reproduce the single-hop replication cycle using a sequence of SRDF and TimeFinder CLI commands.

The following topics provide more information:

- "Manually setting up single hop" on page 340
- "Manually setting up multi-hop" on page 342
- "Setting the symreplicate control parameters" on page 350

Examples To execute the `symreplicate setup` command on a device group (`DevGrp1`) using an options file (`OpFile`):

```
symreplicate -g DevGrp1 setup -options Opfile
```

The first cycle of the `symreplicate start -setup` command puts the devices into the required pair state.

To execute the `symreplicate start` command with the `-setup` option:

```
symreplicate -g DevGrp1 start -options Opfile -setup
```

Using the -exact option

Use the `-exact` option to start the `symreplicate` session with the STD-BCV pair relationships in the exact order that they were associated/added to the device group or composite group.

Using the -optimize option

Use the `-optimize` option in conjunction with the `-setup` option or the `setup` argument to optimize the disk I/O on standard/BCV pairs in the device or composite group.

The `-optimize` option splits all pairs and performs an optimized STD-BCV pairing within the specified group.

If you use the `-optimize` option with device groups, the device pair selection attempts to distribute I/O by pairing devices in the group that are not on the same disk adapter.

Example

```
symreplicate setup -g DgName -optimize
```

Use the `-optimize` option with composite groups to specify the same pairing behavior for an RA group.

Use the `-optimize_rag` option with either the `-setup` option or the `setup` argument to configure pair assignments for RA groups that provide remote I/O optimization (distribution by using different remote disk adapters).

Example: **symreplicate setup -cg *CgName* -optimize_rag**

Note: Single-hop replication does a full optimization on all RA groups.

Using the consistent split option

Use the `-consistent` option with the `start` action to:

- Consistently split all of the BCV pairs on the local array in a typical SRDF configuration
- Consistently split all of the BCV pairs on the Hop 1 remote array in a multi-hop configuration.

Note: This requires a TimeFinder/CG license.

Consistent split operations are automatically retried if the split fails to complete within the allotted window. If a consistent split operation fails due to the consistency timing window closing before the split can complete (`SYMAPI_C_CONSISTENCY_WINDOW_CLOSED`):

- The first-hop local BCV device pairs are automatically resynchronized, and
- The split operation is reattempted.

The consistent split error recovery operation is attempted the number of times specified in the `SYMCLI_REPLICATE_CONS_SPLIT_RETRY` file parameter, defined in the replicate options file.

If a value is not specified, then the recovery operation is attempted 3 times before terminating the `symreplicate` session.

"SYMCLI_REPLICATE_CONS_SPLIT_RETRY=‹NumRetries›" on page 353 provides more information.

Manually setting up single hop

To manually reproduce the single-hop replication cycle using a sequence of SRDF and TimeFinder CLI commands:

1. Wait for any ongoing establish to complete.
2. Split the BCV pairs:

   symmir split -g newdg

3. Establish the SRDF pairs:

   symrdf establish -g newdg -bcv

4. Wait for any ongoing establish to complete.
5. Suspend the SRDF pairs:

   symrdf suspend -g newdg -bcv

6. Establish the BCV pairs:

   symmir establish -g newdg -exact

7. Establish the remote BRBCV pairs:

   symmir establish -g newdg -bcv -rdf -exact

8. Wait for any ongoing establish to complete.
9. Split the remote BRBCV pairs:

```
symmir split -g newdg -bcv -rdf
```

Note: You may have to include additional command options in some of the above steps (for example, `establish -full` for BCV pairs without relationships).

Configuring multi-hop sessions

Figure 119 shows a complete `symreplicate` copy cycle in a multi-hop configuration:

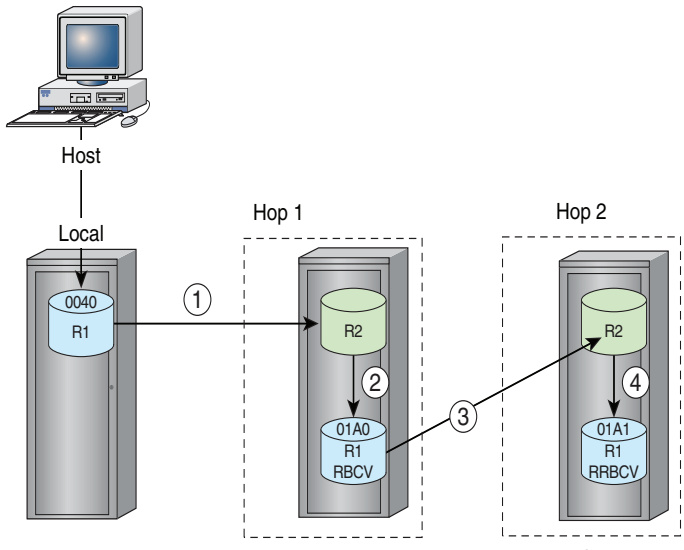


Figure 119 Automated data copy path in multi-hop SRDF

Data copy paths in Figure 119 are:

- **1** - From the local standard device to a standard device on the array at Hop 1
- **2**- From the Hop 1 standard device to its BCV (RBCV)
- **3** - From the RBCV device at Hop 1 to the standard device on the array at Hop 2
- **4** - From the Hop 2 standard device to its BCV (RRBCV)

  Path 2d requires a BCV in the array at Hop 2. The BCV must not be disabled.

Before you begin: setting the hop type and use final parameters

Set the replication type parameter in the replicate options file before you configure a multi-hop `symreplicate` session.

Set the parameter as follows:

```
SYMCLI_REPLICATE_HOP_TYPE=MULTI
```

"SYMCLI_REPLICATE_HOP_TYPE=‹RepType›" on page 350 provides more information.

Set the replication use final BCV parameter in the replicate options file to `FALSE` to prevent the final Hop 2 BCV from being updated:

```
SYMCLI_REPLICATE_USE_FINAL_BCV=FALSE
```

"Setting the symreplicate control parameters" on page 350 provides more information.

Setting up for a multi-hop configuration

To set up a multi-hop `symreplicate` session:

1. Use the `symdg create` command to create an R1 device group (`-g`) or composite group (`-cg`).

   `symdg create newdg2 -type RDF1`

2. Use the `symdg add dev` command to add any number of R1 devices.

   `symdg add dev 0040 -g newdg2 -sid 0001`

3. Use the `symdg add dev` command to remotely associate an equal number of matching sized R1-BCVs or Hop 1 RBCV devices.

   ```
   symbcv associate dev 01A0 -g newdg2 -rdf
   symbcv associate dev 01A1 -g newdg2 -rrdf
   ```

The following must be true before you start a `symreplicate` session without a setup operation:

- The local SRDF pairs must be synchronized
- The BCV pairs must be established
- The remote SRDF pairs must be suspended.
- If the final BCVs in the second-hop array are used, the BCVs must be in the split state.

Device pair state can be configured automatically using the `symreplicate setup` command or the `-setup` option with the `symreplicate start` command.

"Setting up pair states automatically" on page 338 provides more information.

Manually setting up multi-hop

To manually reproduce the multi-hop replication cycle using a sequence of SRDF and TimeFinder CLI commands:

1. Wait for any ongoing establish to complete.
2. Split the BCV pairs (2b in Figure 119):

   `symmir split -g newdg2 -rdf -remote`

   The `-remote` option specifies that the remote SRDF pairs establish.

3. Wait for the establish to complete.
4. Suspend the remote SRDF pairs (2c in Figure 119), and establish the BCV pairs (2b in Figure 119):

   `symmir establish -g newdg2 -rdf -exact`

5. Use either a device file or the `-rrbcv` option to establish the BCV pairs in the second hop (2d in Figure 119):

   `symmir establish -f 2nd_hop_devs.txt -sid SymmID`

or

```
symmir establish -g newdg2 -rrbc
```

Note: To use the `-rrbcv` option, the SRDF BCV devices must have been previously associated with the group, using `symbcv -rrdf`.

6. Wait for any ongoing establish to complete.
7. Split the 2nd hop BCV pairs:

```
symmir split -f 2nd_hop_devs.txt
```

or

```
symmir split -g newdg2 -rrbcv
```

Perform Steps 5 and 7 when you want to use the final hop 2 BCVs in the replicate cycle.

You can use the `-preaction` and `-postaction` options to specify scripts for `symreplicate` to run before and after splitting the BCVs (step 2).

Note: You may have to include additional command options in some of the above steps (such as `establish -full` for BCV pairs without relationships).

Concurrent BCVs with SRDF/AR

Set up concurrent BCVs if you need an independent copy of your data during a replication cycle.

- One BCV copy is associated with the SRDF/AR device group and
- The other BCV copy is not.

  The BCV not associated with the replication cycle receives the same data as the one associated with the SRDF/AR devices. This BCV can be accessed by its host during the `symreplicate` cycle.

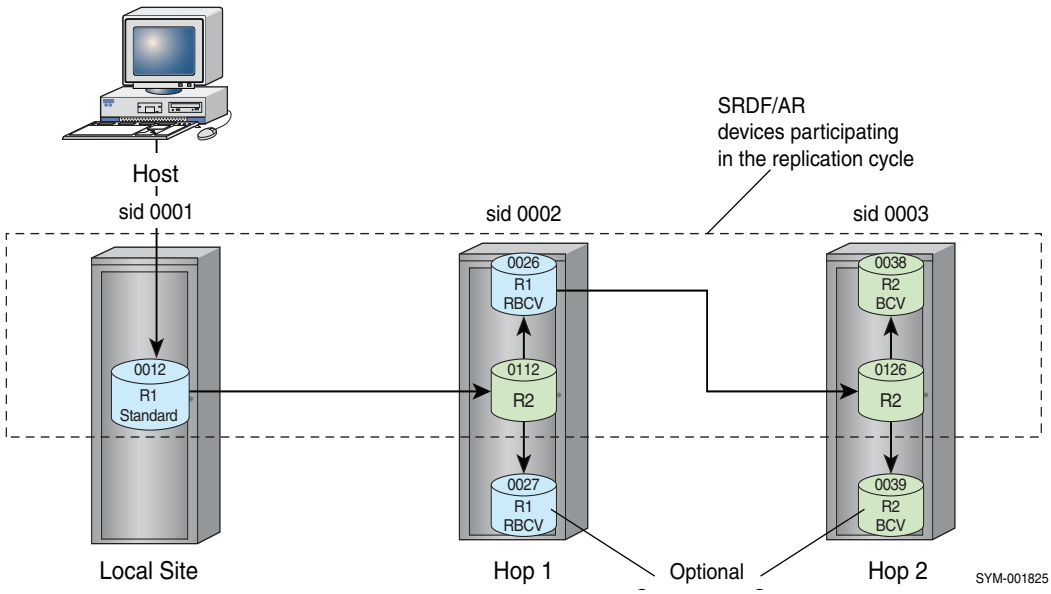


Figure 120 Concurrent BCV in a multi-hop configuration

In Figure 120, Devices 0027 and 0039 are not part of the SRDF/AR copy cycle.

To access these devices from the production host during the SRDF/AR copy cycle, you must define separate device files on the host that include the standard R2 device and the R2 BCV on Hop 1 and Hop 2.

The device files are used to establish the BCV pairs, split BCV pairs, and access the BCV devices.

Setting replication cycle parameters

You can manipulate the replication cycle patterns to fit your needs by setting the following parameters in the symreplicate options file:

`SYMCLI_REPLICATE_CYCLE=`*`CycleTime`*

CycleTime is a timer that indicates the period of time in *minutes* or *hours*:*minutes*(*hh*:*mm*) between when each copy action starts and when it starts again (how often the copy reoccurs). For example, a *CycleTime* of 120 would initiate a new copy every 2 hours.

`SYMCLI_REPLICATE_NUM_CYCLES=`*`NumCycles`*

NumCycles indicates the number of replication cycles (copies) to perform before `symreplicate` exits. For example, a zero value (the default value) results in continuous cycling until the `symreplicate stop` command is issued.

`SYMCLI_REPLICATE_CYCLE_DELAY=`*`Delay`*

Delay indicates the minimum amount of time to wait between the end of one copy cycle and the beginning of the next. For example, a *Delay* of 20 would always force a wait of 20 minutes or more between cycles.

`SYMCLI_REPLICATE_CYCLE_OVERFLOW=`*`OvfMethod`*

OvfMethod indicates how to behave when the actual copy time of your data and/or data transfer throughput is so large as to exceed the *CycleTime* value. Here, the initial copy event has overflowed into the period that should be for the next copy cycle. Possible behavior values are:

`IMMEDIATE` — When overflowed, starts a new cycle immediately after the current copy finishes.

`NEXT` — When overflowed, waits for the copy to finish, and then starts at the next expiration time (*CycleTime*). (Starts the copies on multiples of the *CycleTime* parameter.)

For example, if a 1-hour copy cycle completed in 1.5 hours, the next cycle could be set to begin immediately (`IMMEDIATE`) or in half an hour (`NEXT`).

Setting the first time cycle parameters

You may not have enough information to set the exact cycle time parameters when you first create the SRDF configuration.

Best practice

- Start the `symreplicate` session with the basic parameters set.
- Use `symreplicate query` to monitor session progress, and record the timing results of the initial copies.
- Adjust the various timing parameters to best accommodate the copy requirements for your needs.

Table 40 provides two parameter setups for an initial `symreplicate` session trial:

Table 40 Initial setups for cycle timing parameters

SYMCLI_REPLICATE_CYCLE=60 SYMCLI_REPLICATE_CYCLE_DELAY=0 SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT	Every hour if possible, or every 2, or 3 hours based on data throughput and size.
SYMCLI_REPLICATE_CYCLE=0 SYMCLI_REPLICATE_CYCLE_DELAY=60	Cycle through the first copy, then wait 60 minutes (delay), and then another cycle, delay, and so on.

Viewing cycle time and invalid track statistics

Use the `symreplicate stats` command to display statistical information for cycle time and invalid tracks.

Use the command to display cycle time and invalid tracks for a specified:

- Device group (`-g`)
- Composite group (`-cg`)
- Symmetrix ID (`-sid`)

Use the `-log` option to write information to a specified log file.

Use the `-cycle` option to display only cycle time statistics for the last SRDF/AR cycle time, the maximum cycle time and the average cycle time.

Use the `-itrks` option to display only invalid track statistics for the last SRDF/AR cycle, the maximum invalid tracks and the average number of invalid tracks per SRDF/AR cycle.

Use the `-all` option (default) to display both the cycle time and invalid tracks statistics.

Example To display both cycle time and invalid track statistics for device group `srdfar` on SID 1123:

```
symreplicate -g srdfar -sid 123 -all stats

Group Name: srdfar

Cycle Time (hh.mm.ss):
---------------------------------------

Last Cycle Time:  06:10:01
Max Cycle Time:   08:00:00
Avg Cycle time:   06:00:00

Invalid Tracks:
---------------------------------------

Last Cycle:  12345 ( 9055.5 MB)
Maximum:     10780 ( 8502.3 MB)
Average:     11562 ( 7500.0 MB)
```

Logging symreplicate steps

To track the steps in a `symreplicate` session, set the log step entry in the options file to `TRUE`:

```
SYMCLI_REPLICATE_LOG_STEP=TRUE
```

When this option is enabled, `symreplicate` writes an entry to the SYMAPI log file after each step is completed.

Log entries contain the time that the step ended and whether it was successful.

"SYMCLI_REPLICATE_LOG_STEP=‹TRUE|FALSE›" on page 351 provides more information.

Clustered SRDF/AR

Clustered SRDF/AR enables you to start, stop, and restart `symreplicate` sessions from any host connected to any local array participating in the `symreplicate` session.

In the clustered SRDF/AR environment, you can write the replication log file directly to the Symmetrix File System (SFS) instead of the local host directory of the node that began the session.

If the primary node should fail, then any locally attached host to the array containing the log file can restart the SRDF/AR session from where it left off.

Writing log files to a specified SFS

Use the `symreplicate start` command with the `-sid` and `-log` options to write the log file to the SFS. You must specify:

- `-sid` – ID of the array where the log file is to be stored at the start of the `symreplicate` session.

- `-g` or `-cg` - Group name.
- `-log` *`LogFilename`* - (Optional) User log filename.

Usage notes

- If you do not specify the Symmetrix ID (`-sid`) at the start of the session, the log file is written to local disk using the default SYMAPI log directory. This is not restartable from another node.
- If you begin a session and specify a user log file name (`-log` *`LogFilename`*), you must specify the `-log` option for all other commands in the session sequence.
- If you begin a session and specify only the group name (`-g`, `-cg`):
  - The log file is given the same name as the group,
  - You must specify only the `-g` or `-cg` option for all other commands in the session sequence.

HYPERMAX OS

- In HYPERMAX OS/Solutions Enabler 8.0.1 or higher, the following options for the `symreplicate start` command are not supported, and the command fails with the message “Illegal option”.
  - - vxfs
  - -rdb

To write the log file for device group `session1` to a file named `srdfar1.log` at the SFS on array 201:

```
symreplicate start -g session1 -log srdfar1.log -sid 201
```

Restarting from another host

When log files are sent to the SFS, then any locally attached host to the array containing the log file can restart the SRDF/AR session from where it left off.

Use the `symreplicate restart` command with the `-recover` option to restart the session using the specified log and recover the device locks from the previous session.

Command usage notes

The `-recover` option recovers the device locks from the previously started session. Verify that no other currently running `symreplicate` session is using the same devices before using the `-recover` option.

You do not need to specify the device or composite group name (`-g`, `-cg`) on the host where the session is restarted.

Example

To restart the SRDF/AR session from another local host:

```
symreplicate restart -g session1 -log srdfar1.log -sid 201 -recover
```

Displaying log files written to the SFS

symreplicate list

Use the `symreplicate list` command with the `-sid` option to display a list of the current SRDF/AR log files written to the SFS at the specified SID.

Use the `symreplicate list` command with the `-sort` option to sort the log file list by name (default) or type.

Example To list the log files at SID 201:

```
symreplicate list -sid 201
```

symreplicate show

Use the `symreplicate show` command to display the information content of a particular log file.

Usage notes

- Log filename (`-log`) and the Symmetrix ID (`-sid`) are required.
- Use the `-args` option to display only command line arguments.
- Use the `-devs` option to display only devices.
- Use the `-opts` option to display only options.
- Use the `-all` option (default) to display all available information contained in the log.

EMC Solutions Enabler CLI Command Reference provides more information.

Example To display the log file `srdfar1.log` at SID 201:

```
symreplicate show -log srdfar1.log -sid 201 -all
```

Deleting a log file written to SFS

Use the `symreplicate delete -log` command to delete the specified log file written to SFS.

Usage notes

Specify either the group name (`-g`, `-cg`) or the log filename (`-log`) depending on whether you defined a user log name when you began the session.

Example To delete log file `srdfar1.log` written to the SFS:

```
symreplicate delete -log srdfar1.log
```

Setting symreplicate parameters in the options file

You can modify parameters in the `symreplicate` options file to:

- Set replication retry and sleep timers
- Control replicate behavior

Note: If you specify an options file on `restart`, you may not change the following options:
"SYMCLI_REPLICATE_USE_FINAL_BCV=‹TRUE|FALSE›"
"SYMCLI_REPLICATE_HOP_TYPE=‹RepType›"
If you attempt to change these options, an error message is displayed. All other options may be changed, and the new values take effect immediately.

IMPORTANT

You must specify the *RepType*. See:

"SYMCLI_REPLICATE_HOP_TYPE=‹RepType›" on page 350.

Set a nonzero value for either a *CycleTime* or a *Delay* time, (even though their default values are zero). See:

"SYMCLI_REPLICATE_CYCLE=CycleTime" on page 344

"SYMCLI_REPLICATE_CYCLE_DELAY=Delay" on page 344

Option file format

Make sure that your changes conform to the syntax in the example below.

The desired value is entered for the italicized text.

Lines beginning with a "#" (comment) are ignored by SYMCLI:

```
#Comment
SYMCLI_REPLICATE_HOP_TYPE=<RepType>
SYMCLI_REPLICATE_CYCLE=<CycleTime>
SYMCLI_REPLICATE_CYCLE_OVERFLOW=<OvfMethod>
SYMCLI_REPLICATE_CYCLE_DELAY=<Delay>
SYMCLI_REPLICATE_NUM_CYCLES=<NumCycles>
SYMCLI_REPLICATE_USE_FINAL_BCV=<TRUE|FALSE>
SYMCLI_REPLICATE_LOG_STEP=<TRUE|FALSE>
SYMCLI_REPLICATE_GEN_TIME_LIMIT=<TimeLimit>
SYMCLI_REPLICATE_GEN_SLEEP_TIME=<SleepTime>
SYMCLI_REPLICATE_RDF_TIME_LIMIT=<TimeLimit>
SYMCLI_REPLICATE_RDF_SLEEP_TIME=<SleepTime>
SYMCLI_REPLICATE_BCV_TIME_LIMIT=<TimeLimit>
SYMCLI_REPLICATE_BCV_SLEEP_TIME=<SleepTime>
SYMCLI_REPLICATE_MAX_BCV_SLEEP_TIME_FACTOR=<Factor>
SYMCLI_REPLICATE_MAX_RDF_SLEEP_TIME_FACTOR=<Factor>
SYMCLI_REPLICATE_PROTECT_BCVS=<Protection>
SYMCLI_REPLICATE_TF_CLONE_EMULATION=<TRUE|FALSE>
SYMCLI_REPLICATE_PERSISTENT_LOCKS=<TRUE|FALSE>
SYMCLI_REPLICATE_CONS_SPLIT_RETRY=<NumRetries>
SYMCLI_REPLICATE_R1_BCV_EST_TYPE=<EstablishType>
SYMCLI_REPLICATE_R1_BCV_DELAY=<EstablishDelay>
SYMCLI_REPLICATE_FINAL_BCV_EST_TYPE=<EstablishType>
SYMCLI_REPLICATE_FINAL_BCV_DELAY=<EstablishDelay>
SYMCLI_REPLICATE_ENABLE_STATS=<TRUE|FALSE>
SYMCLI_REPLICATE_STATS_RESET_ON_RESTART=<TRUE|FALSE>
```

Setting replication retry and sleep times

Control how long and how often `symreplicate` executes control operations by setting the following parameters in the `symreplicate` options file:

`SYMCLI_REPLICATE_GEN_TIME_LIMIT=`*`TimeLimit`*

Controls how long errors of a general nature, such as waiting for a lock, are retried.

`SYMCLI_REPLICATE_RDF_TIME_LIMIT=`*`TimeLimit`*

Controls how long to wait for SRDF devices to enter a specific state.

`SYMCLI_REPLICATE_BCV_TIME_LIMIT=`*`TimeLimit`*

Controls how long to wait for BCV devices to enter a specific state.

`SYMCLI_REPLICATE_GEN_SLEEP_TIME=`*`SleepTime`*

Controls how long `symreplicate` should sleep before retrying a general operation.

`SYMCLI_REPLICATE_RDF_SLEEP_TIME=`*`SleepTime`*

Controls the minimum time `symreplicate` should sleep before retrying an SRDF operation.

`SYMCLI_REPLICATE_BCV_SLEEP_TIME=`*`SleepTime`*

Controls the minimum time `symreplicate` should sleep before retrying a BCV operation.

`SYMCLI_REPLICATE_MAX_BCV_SLEEP_TIME_FACTOR=`*`Factor`*

Controls the maximum time that `symreplicate` sleeps before checking the BCV device state.

`SYMCLI_REPLICATE_MAX_RDF_SLEEP_TIME_FACTOR=`*`Factor`*

Controls the maximum time that `symreplicate` sleeps before checking the SRDF device state.

Setting the symreplicate control parameters

You can modify the following parameters in the `symreplicate` options file to control replicate behavior:

`SYMCLI_REPLICATE_HOP_TYPE=<`*`RepType`*`>`

Defines your configured environment in which to operate the data `symreplicate` session. This parameter is not optional and must be specified. Possible *RepType* values are:

`SINGLE` — Single-hop configuration

`MULTI` —Multi-hop configuration

`SYMCLI_REPLICATE_USE_FINAL_BCV=<`*`TRUE`*`|`*`FALSE`*`>`

Indicates whether to update the BCV in the final (last) remote array (for multi-hop only).

`TRUE` –(default) Replicates data copy the BCV in the final (last) remote array.

`FALSE` -The second hop BCV devices will be omitted.

```
SYMCLI_REPLICATE_PROTECT_BCVS=
          <NONE|BOTH|LOCAL|REMOTE|FIRST_HOP|SECOND_HOP>
```

`NONE` - (default) Establishes BCV-STD pairs without the protective establish behavior, relating to two-way mirrored BCV devices.

`LOCAL` or `REMOTE` - Causes the two mirrors of the BCV to be moved or joined to the standard device.

`BOTH` - Both the local BCV mirrors and the remote BCV mirrors get joined to their standard device.

`FIRST_HOP` or `SECOND_HOP` - Performs the protect BCV establish for first or second hop devices only in a multi-hop configuration.

`SYMCLI_REPLICATE_CYCLE=<CycleTime>`

Defines the period to wait between copy operations in total *minutes* or in an *hours*:*minutes* (*hh:mm*) format.

`SYMCLI_REPLICATE_CYCLE_DELAY=<Delay>`

Specifies the minimum time to wait between adjacent cycles. Even if a cycle overruns the specified *CycleTime* and *OvfMethod* is set to `IMMEDIATE` when *Delay* is specified, the session waits this delay time before beginning another cycle.

`SYMCLI_REPLICATE_NUM_CYCLES=<NumCycles>`

Specifies the number of cycles to perform before exiting. If you specify a value of zero, the `symreplicate` session cycles forever. The *NumCycles* default value is zero.

`SYMCLI_REPLICATE_CYCLE_OVERFLOW=<OvfMethod>`

Describes what to do if the cycle overruns the specified *CycleTime*. Possible *OvfMethod* values are:

`IMMEDIATE` — Begins next cycle immediately (the default)

`NEXT` — Skips this copy cycle and wait for the next to begin

`SYMCLI_REPLICATE_LOG_STEP=<TRUE|FALSE>`

`TRUE` - Writes a log entry to the SYMAPI log file after each step of the `symreplicate` cycle is completed. The entry displays the time that the step ended and whether the step was successful.

`SYMCLI_REPLICATE_GEN_TIME_LIMIT=<TimeLimit>`

Indicates how long errors of a general nature should be retried (for example, attempting to acquire a array lock). Currently, the general *TimeLimit* only applies when initiating an SRDF split or establish operation.

The default general *TimeLimit* is 00:30 if not specified.

The *TimeLimit* value enables you to control how long `symreplicate` retries certain types of operations. *TimeLimit* must be specified using one of the following formats:

hh:mm — Specifies the number of hours and minutes

sss — Specifies the number of seconds

A *TimeLimit* specified as zero (0) indicates that no time limit applies, causing the operation to be retried indefinitely.

`SYMCLI_REPLICATE_RDF_TIME_LIMIT=<TimeLimit>`

Indicates how long to wait for SRDF devices to enter a specific state. For example, after successfully issuing the command to establish an R2 BCV device with the corresponding R1 standard device, `symreplicate` waits the indicated length of time for the devices to become synchronized.

The default SRDF *TimeLimit* is 04:00 if not specified.

`SYMCLI_REPLICATE_BCV_TIME_LIMIT=<TimeLimit>`

Indicates how long to wait for BCV devices to enter a specific state. For example, after successfully issuing the command to establish a BCV device with the corresponding standard device, `symreplicate` waits the indicated length of time for the devices to become synchronized.

The default BCV *TimeLimit* is 02:00 if not specified.

`SYMCLI_REPLICATE_GEN_SLEEP_TIME=<SleepTime>`

Indicates how long `symreplicate` should sleep before retrying a general operation (for example, attempting to acquire a array lock). Currently, the general *SleepTime* only applies when initiating an SRDF split or establish operation. The default general *SleepTime* is 10 seconds if not specified.

The *SleepTime* value enables you to control how long `symreplicate` sleeps before retrying certain types of operations. *SleepTime* must be specified using one of the following formats:

hh:mm — Specifies the number of hours and minutes

sss — Specifies the number of seconds

A *SleepTime* must be specified as greater than zero (0).

`SYMCLI_REPLICATE_RDF_SLEEP_TIME=<SleepTime>`

Indicates the minimum length of time that `symreplicate` should sleep before retrying an SRDF device operation. For example, after issuing the command to establish an R2 BCV device with the corresponding R1 standard device, `symreplicate` sleeps the indicated length of time before retrying the operation.

The default SRDF *SleepTime* is 15 seconds if not specified.

`SYMCLI_REPLICATE_BCV_SLEEP_TIME=<SleepTime>`

Indicates the minimum length of time that symreplicate should sleep before retrying a BCV device operation. For example, after issuing the command to establish a BCV device with the corresponding standard device, `symreplicate` sleeps the indicated length of time before retrying the operation.

The default BCV *SleepTime* is 10 seconds if not specified.

`SYMCLI_REPLICATE_MAX_BCV_SLEEP_TIME_FACTOR=<Factor>`

Provides a way to specify the maximum time that `symreplicate` sleeps before checking again to see if BCV devices have entered a specific state. The product of this value multiplied by the sleep time gives the maximum time that `symreplicate` sleeps. The factor is specified using a nonzero integer. If not specified, the default factor is 3.

By default, `symreplicate` sleeps between 10 and 30 seconds when checking on the state of BCV devices, up to a maximum time of 2 hours.

`SYMCLI_REPLICATE_MAX_RDF_SLEEP_TIME_FACTOR=<Factor>`

Provides a way to specify the maximum time that `symreplicate` sleeps before checking again to see if SRDF devices have entered a specific state. The product of this value multiplied by the sleep time gives the maximum time that `symreplicate` sleeps. The factor is specified using a nonzero integer.

If not specified, the default factor is 4.

Note: By default, `symreplicate` sleeps between 15 and 60 seconds when checking on the state of SRDF devices, up to a maximum time of 4 hours.

`SYMCLI_REPLICATE_TF_CLONE_EMULATION=<TRUE|FALSE>`

Indicates that TF/Clone emulation is enabled/disabled.

`FALSE` – (default) The TF/Clone emulation default is disabled.

`TRUE` - Indicates that clone emulation is enabled.

`SYMCLI_REPLICATE_PERSISTENT_LOCKS=<TRUE|FALSE>`

Allows device locks to persist in the event of a system crash or component failure.

`TRUE` - Causes symreplicate to acquire the device locks for the `symreplicate` session with the `SYMAPI_DLOCK_FLAG_PERSISTENT` attribute.

`FALSE` - The persistent attribute will not be used to acquire the device locks for the session. If the base daemon (storapi daemon) is running and persistent locks are not set, the base daemon will release the device locks in the event of a failure.

`SYMCLI_REPLICATE_CONS_SPLIT_RETRY=<NumRetries>`

Specifies the number of error recovery attempts that will be made when a consistent split operation fails because the timing window closed before the split operation completed. A default retry value of 3 will be used if the SYMCLI_REPLICATE_CONS_SPLIT_RETRY option parameter is not specified when a consistent split (`-consistent`) is requested. A retry value of 0 indicates that no retry attempts should be made.

`SYMCLI_REPLICATE_R1_BCV_EST_TYPE=<EstablishType>`

Specifies the establish type for the local/first hop BCV devices. *EstablishType* specifies the way that BCV establish operations will be executed by TimeFinder. One of the following values may be specified:

`SINGULAR` — BCV devices will be established one at a time; the next device will not be established until the previous device has been established.

`SERIAL` — BCV devices will be established as fast as the establish requests can be accepted by the array.

`PARALLEL` — BCV devices establish requests will be passed in parallel to each of the servicing DA directors.

`SYMCLI_REPLICATE_R1_BCV_DELAY=<EstablishDelay>`

Denotes how long to wait between issuing establish requests. Establish types of SINGULAR and PARALLEL, for an ‹*EstablishDelay*› can be specified through the SYMCLI_REPLICATE_R1_BCV_DELAY file parameter.

SYMCLI_REPLICATE_FINAL_BCV_EST_TYPE=<*EstablishType*>

Identifies the establish type for the remote/second hop BCV devices.

SYMCLI_REPLICATE_FINAL_BCV_DELAY=<*EstablishDelay*>

Indicates how long to wait between issuing establish requests for the remote/second hop BCV devices. For an establish type of PARALLEL the delay value indicates how long to wait before passing the next establish request to an individual servicing DA director. An establish delay of 0 to 30 seconds may be specified with a value of 0 being the default.

SYMCLI_REPLICATE_ENABLE_STATS=<*TRUE*|*FALSE*>

Enables or disables the gathering of statistics.

TRUE - (default) Indicates that statistics gathering is enabled.

FALSE - Indicates that statistics gathering is to be disabled.

SYMCLI_REPLICATE_STATS_RESET_ON_RESTART=<*TRUE*|*FALSE*>

Resets statistics when a restart action is executed.

TRUE - Indicates that statistics are to be reset when restarting a symreplicate session.

FALSE - (default) Statistics are not reset upon restart of a symreplicate session.

Managing locked devices

Device external locks in the array are held during the entire symreplicate session. Device external locks block other applications from altering device states while the symreplicate session executes.

When a symreplicate session terminates because the SRDF link goes down unexpectedly, the locked devices prevent session restart when the SRDF link is restored.

You can recover, release or acquired to persist device locks.

Recovering locks

Use the symreplicate start or restart command with the -recover option to recover the device locks and restart the session.

Note: Device locks can be recovered as long as exactly the same devices are still locked under the lock holder ID of the previous symreplicate session.

Releasing locks

Optionally, you can release the device external locks held in the array for a terminated SRDF/AR session.

Locks may need to be released manually if a session is terminated unexpectedly due to a system crash or component failure. Device locks for a terminated session can be released manually for a device group, composite group or log file without restarting the session.

Use the `symreplicate release` command to release any device external locks associated with devices in the specified device group that are still held from when they were locked from the terminated SRDF/AR session.

Example To release devices locks on a terminated session for device group `prod` on array `35002`:

```
symreplicate -g prod release -sid 35002
```

Restrictions to release locks

- The SRDF/AR session for the targeted devices must not be active.
- Devices must have been locked by the previous session and the lock holder ID must match the previous session's ID.
- The number of devices to be unlocked must be less than or equal to the total number of devices in the previous SRDF/AR session.

The force (`-force`) option is required to release device locks in the following situations:

- If the release action is requested in a clustered SRDF/AR environment on a host that did not initiate the session and the status of the session cannot be determined.
- If any of the devices' lock holder ID in the targeted SRDF/AR session do not match the session's lock hoder ID, and the user wants to release the devices locked with the session's lock holder ID.
- If the lock holder ID for some devices in the targeted SRDF/AR session do not match the lock holder ID of that session, and the user wants to release the devices locked with the session's original lock holder ID.

Acquiring persistent locks

If the base daemon (SYMAPI daemon) is running, device locks are automatically released in the event of a system crash or component failure.

To acquire the device using the persistent attribute, set the persistent locks parameter in the symreplicate options file to `TRUE`:

```
SYMCLI_REPLICATE_PERSISTENT_LOCKS=TRUE
```

See "SYMCLI_REPLICATE_PERSISTENT_LOCKS=‹TRUE|FALSE›" on page 353.

CHAPTER 11
TimeFinder and SRDF operations

This chapter describes the following topics:

- TimeFinder consistent splits in SRDF configurations 358
  - "Enginuity Consistency Assist"
- Multi-hop operations 359
  - "Before you begin: system-wide device groups"
  - "Controlling basic operations in a multi-hop configuration"
  - "System-wide splits"
- TimeFinder SnapVX and SRDF 363
  - "TimeFinder SnapVX and Cascaded SRDF"
  - "TimeFinder SnapVX and Concurrent SRDF"

TimeFinder consistent splits in SRDF configurations

TimeFinder consistent split allows you to split off a consistent, restartable copy of a database management array within seconds with no service interruption.

A concurrent split helps to avoid inconsistencies and restart problems that can occur when splitting database-related BCVs without first quiescing the database.

Consistent split operations are implemented using the Enginuity Consistency Assist (ECA) feature. ECA allows you to consistently activate copy sessions across multiple heterogeneous hosts.

Consistent split operations can also be used in conjunction with SRDF Automated Replication (SRDF/AR) to set up automatic remote mirroring according to a predefined copy schedule.

SRDF/AR is explained in "SRDF/Automated Replication" on page 335.

Enginuity Consistency Assist

TimeFinder consistent split operations are performed using the `symir` command with `-consistent` option.

You can also use the `symreplicate` command with the `-consistent` option to run a copy cycle that freezes I/O to all devices in a device or composite group for both single-hop and multi-hop configurations.

Figure 121 on page 359 depicts how a control host can perform ECA consistent splits for three database hosts that access devices on a array.

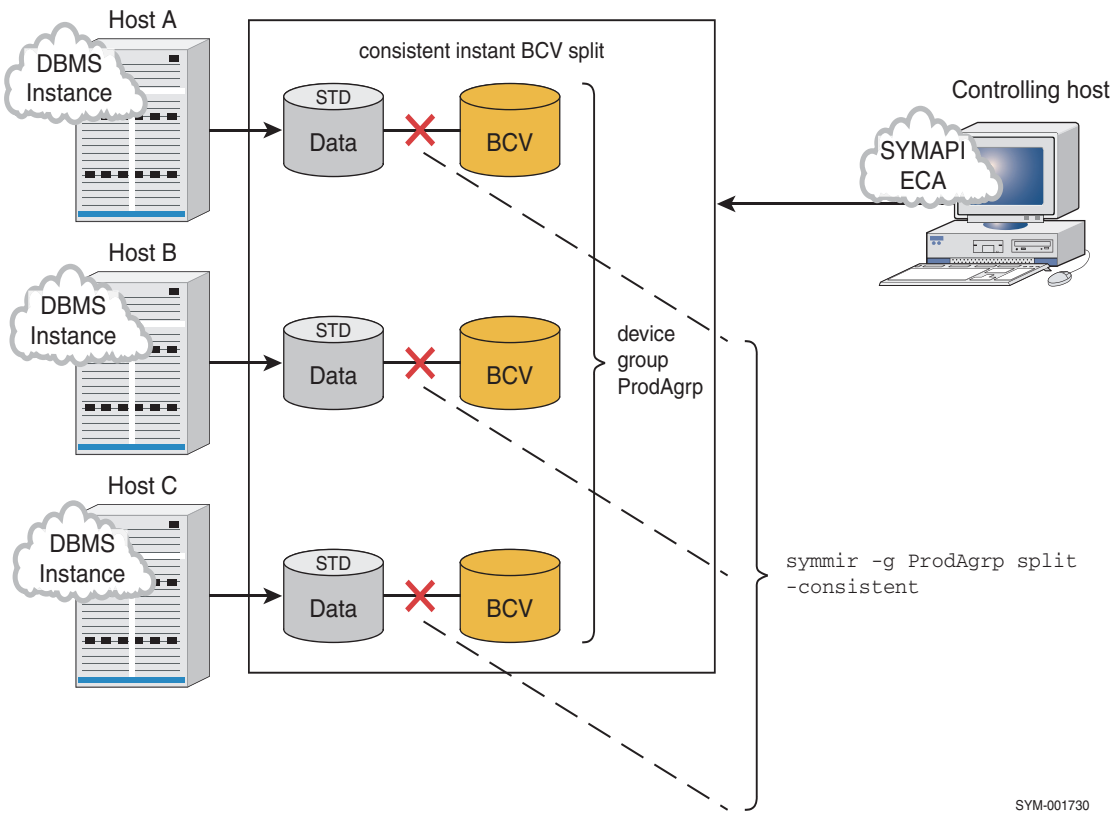


Figure 121 ECA consistent split

To consistently split BCV pairs using ECA:

- You must have either a control host with no database or a database host with a dedicated channel.
- Device or composite groups must be created on the controlling host for the target database to be consistently split.

  These groups can be created to include all of the devices being accessed or defined by database host access.

For example, if you define a device group that includes all of the devices being accessed by Hosts A, B, and C, then you can consistently split all of the BCV pairs related to those hosts with a single command.

Multi-hop operations

You can manage various compounded remote configurations using both the TimeFinder and SRDF components of SYMCLI.

Figure 123 on page 361, shows multiple sites (remote Sites B and C) remotely mirroring to a local array at Site A.

The most typical configuration is a remote site (Site B in Figure 123), functioning as a remote mirror to standard devices (Site A).

A third site (Site C) can remotely mirror just the BCV devices at Site A.

Multi-hop SRDF

You can also configure a multi-hop to a second-level SRDF.

In Figure 123:

- Site D remotely mirrors standard devices at Site A, and
- Site E remotely mirrors Site A's BCV.
- The `symrdf` command manages the SRDF pairs within the SRDF link
- The `symmir` command manages the BCV pairs within any one site.

Before you begin: system-wide device groups

`symmir` operations require an existing group of SRDF devices.

To create a device group containing STD and BCV RDF1 devices:

1. Use the `symdg create` command to create an empty device group:

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

2. Use the `symdg add dev` command to add devices to the new device group:

   ```
   symdg -g prod add dev 0001 -sid 344402 DEV001
   ```

3. Use the `symbcv associate` commands to associate the devices with a local BCV, and remote BCVs:

```
symbcv -g prod associate dev 000A BCV001
symbcv -g prod associate dev 000C -rdf RBCV001
symbcv -g prod associate dev 0009 -bcv -rdf BRBCV001
symbcv -g prod associate dev 0004 -rrdf RRBCV001
```

All devices must be established with the `symmir` and `symrdf` commands.

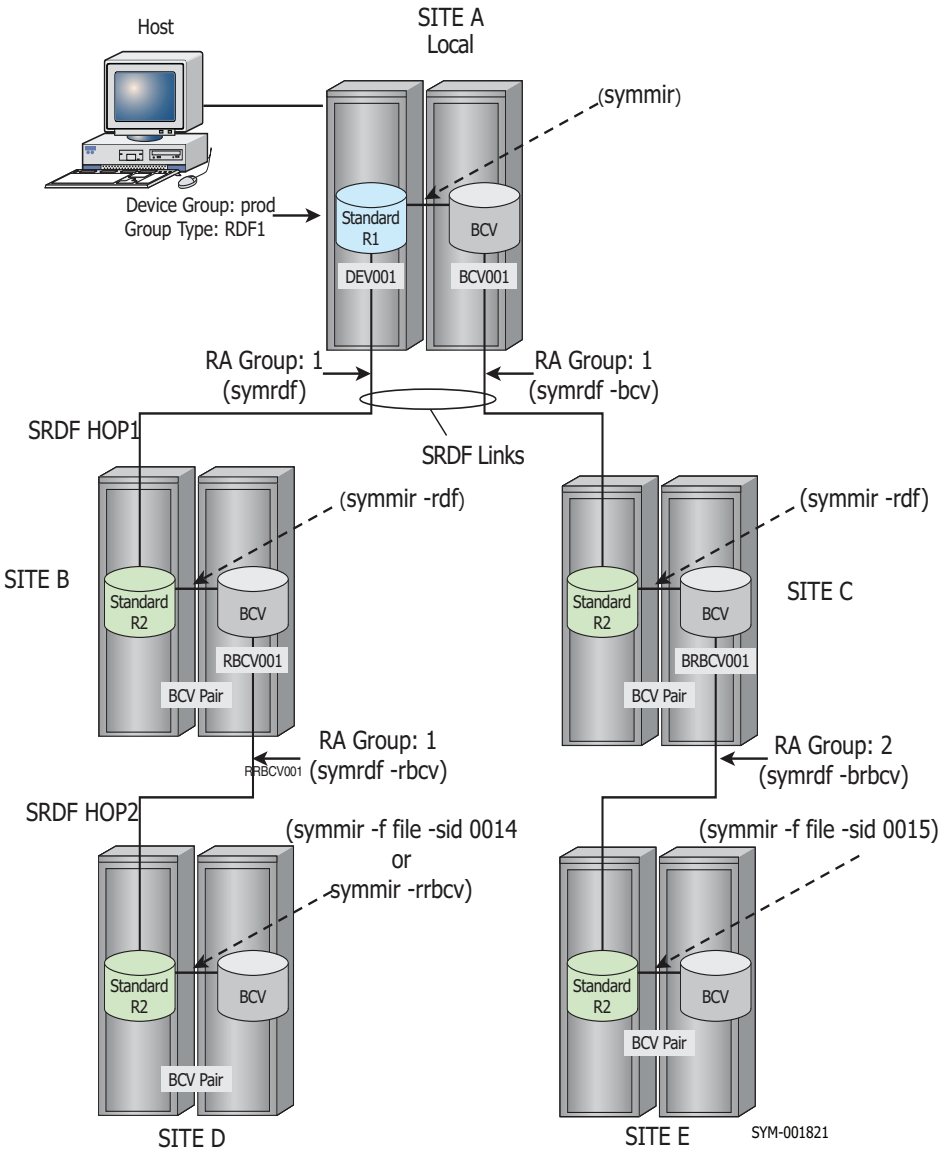


Figure 122 Commands used to perform splits in a complex configuration

Controlling basic operations in a multi-hop configuration

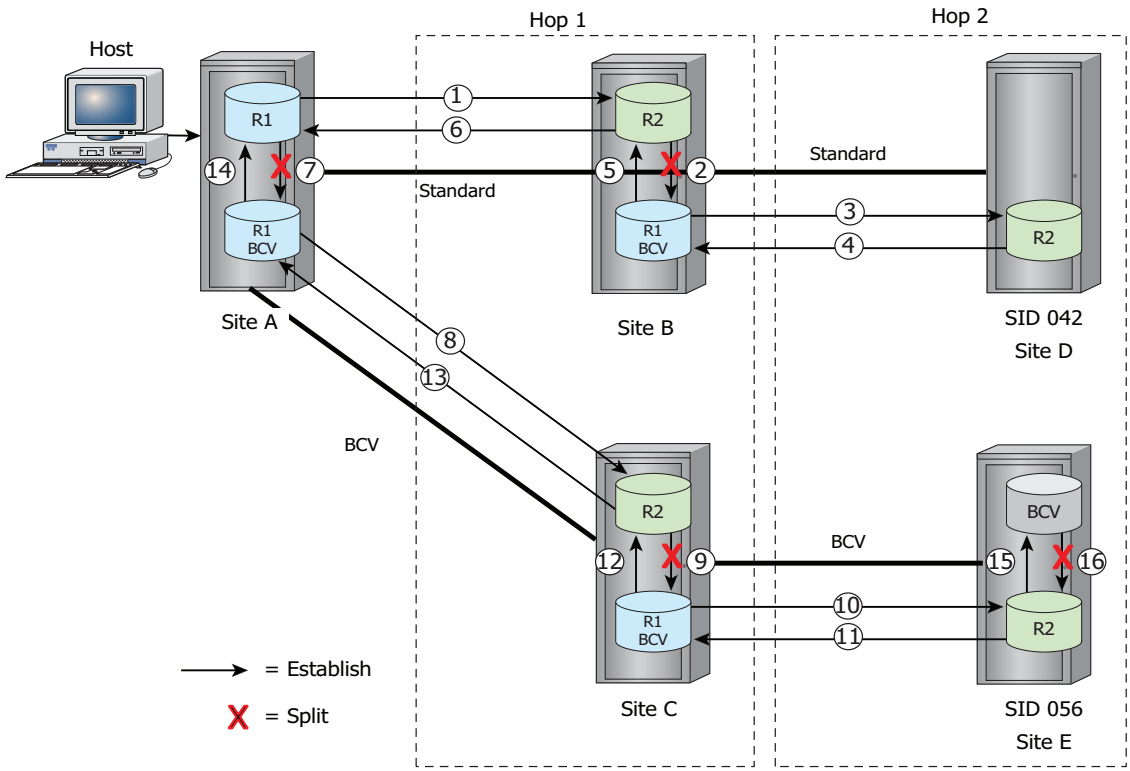


Figure 123 Basic operations in multi-hop SRDF configurations

Table 41 is a sequence of commands to perform basic control operations in a multi-hop configuration.

Each Step number correlates to a bubble number in Figure 123.

Table 41 Basic operations in a multi-hop configuration (page 1 of 2)

Step	CLI control operation	Description
1	symrdf -g ‹› establish	Creates the standard associated hop 1 copy.
2	symmir -g ‹› split -rdf	Splits the standard associated hop 1 BCV device pair.
3	symrdf -g ‹› establish -rbcv	Creates the standard associated hop 2 copy.
4	symrdf -g ‹› restore -rbcv	Restores the standard associated hop 1 BCV with the hop 2 copy.
5	symmir -g ‹› restore -rdf	Restores the standard associated hop 1 copy with the hop 1 BCV.
6	symrdf -g ‹› restore	Restores the standard device with the hop 1 copy.
7	symmir -g ‹› split	Splits the standard/BCV pair.
8	symrdf -g ‹› establish -bcv	Creates the BCV associated hop 1 remote copy.
9	symmir -g ‹› split -rdf -bcv	Splits the BCV associated hop 1 device pair.
10	symrdf -g ‹› establish -brbcv	Creates the BCV associated hop 2 copy.
11	symrdf -g ‹› restore -brbcv	Restores the BCV associated hop 1 BCV with the hop 2 copy.

Table 41 Basic operations in a multi-hop configuration (page 2 of 2)

Step	CLI control operation	Description
12	symmir -g ‹› restore -rdf -bcv	Restores the standard device associated hop 1 copy with the hop 1 BCV.
13	symrdf -g ‹› restore -bcv	Restores the BCV device with the hop 1 copy.
14	symmir -g ‹› restore	Restores the standard device with the BCV copy.
15	symmir -f ‹› -sid 056 establish or symmir -g ‹› -rrbcv establish	Creates the BCV associated hop 2 BCV copy.
16	symmir -f ‹› -sid 056 split or symmir -g ‹› -rrbcv	Splits the BCV-associated hop 2 device pair.

System-wide splits

Figure 123 shows how the `symmir` and `symrdf` commands might be applied to split operations in a complex configuration.

Note: You must have established SRDF device groups before you perform any `symmir` and `symrdf` operations.

Examples

To split the BCV pair within Site A:

```
symmir -g prod split
```

To split SRDF pairs at Site B from host-connected Site A:

```
symrdf -g prod split
```

To split the BCV pairs within Site B:

```
symmir -g prod -rdf split
```

To split BCV SRDF pairs at Site C from host-connected Site A:

```
symrdf -g prod -bcv split
```

To split the BCV pairs within Site C:

```
symmir -g prod -rdf -bcv split
```

To split BCV SRDF pairs at Site D from host standard-associated Site B:

```
symrdf -g prod -rbcv split
```

To split the BCV pairs within Site D:

```
symmir -f dfile -sid 0014 split
```

or

```
symmir -g prod -rrbcv split
```

To split BCV SRDF pairs at Site E from host BCV-associated Site C:

```
symrdf -g prod -brbcv split
```

To split the BCV pairs within Site E (hop 2):

```
symmir -f dfile -sid 0015 split
```

Perform operations such as `establish` and `restore` in the same manner for remote sites.

EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide provides more information.

TimeFinder SnapVX and SRDF

HYPERMAX OS introduces TimeFinder SnapVX. SnapVX creates snapshots by storing changed tracks directly in the Storage Resource Pool of the source device. With SnapVX:

- You do not need to specify a target device and source/target pairs when you create a snapshot.
- You can create links from the snapshot to one or more target devices.
- You can link and relink until the correct snapshot is located.

This section provides an example of:

- "TimeFinder SnapVX and Cascaded SRDF"
- "TimeFinder SnapVX and Concurrent SRDF"

IMPORTANT

Starting in Starting in Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR you can manage SRDF operations using storage groups. "Storage groups" on page 26 provides more information.

HYPERMAX OS uses emulations to transparently convert legacy commands (TimeFinder/Clone, TimeFinder VP Snap, and TimeFinder/Mirror) to SnapVX commands. You can still run existing scripts that include legacy commands, but the underlying mechanism is SnapVX.

EMC VMAX3 Family Product Guide provides detailed information about TimeFinder SnapVX.

TimeFinder SnapVX and Cascaded SRDF

The following example uses device groups to manage TimeFinder operations in a cascaded SRDF configuration.

1. Create device group DeptAB, add local devices 00019:0001A as R1 (source) devices, and 0001D:0001E as TGT devices:

   ```
   symdg create DeptAB -type ANY
   symdg -g DeptAB -sid 000197300076 addall dev -devs 00019:0001A
   symdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt
   ```

2. Add devices 0001D:0001E on remote array (R21, 1st hop) as TGT devices.

   ```
   symdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt
   -rdf
   ```

3. Add devices 0001D:0001E in RDF group 12 on remote array (R2, 2nd hop) as TGT devices:

```
symdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt
-rdf
```

Figure 124 on page 364 shows the resulting configuration:

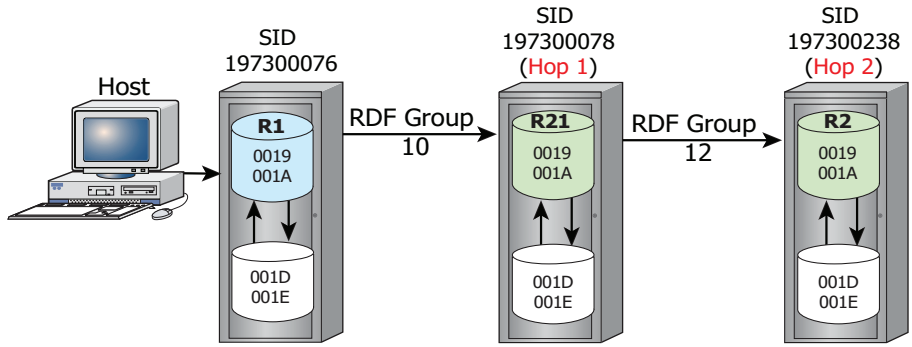


Figure 124 SnapVX and Cascaded SRDF

The following examples use the configuration shown in Figure 124 on page 364:

- Create, activate, and link a SnapVX snapshot (named LocalSnap) on the local array:

```
symsnapvx -g DeptAB establish -name LocalSnap
symsnapvx -g DeptAB -snapshot_name LocalSnap link
```

- Create, activate, and link a SnapVX snapshot (named Hop1Snap) on the remote array at Hop 1:

```
symsnapvx -g DeptAB establish -name Hop1Snap -rdf
symsnapvx -g DeptAB -snapshot_name Hop1Snap link -rdf
```

- Create, activate, and link a SnapVX snapshot (named Hop2Snap) on the remote array at Hop 2:

```
symsnapvx -g DeptAB establish -name Hop2Snap -hop2
symsnapvx -g DeptAB -snapshot_name Hop2Snap link -hop2
```

TimeFinder SnapVX and Concurrent SRDF

The following example uses composite groups to manage TimeFinder operations in a concurrent SRDF configuration.

1. Create composite group DeptPR and with RDF consistency enabled, add devices in RDF group 20 as source (R11) devices:

```
symcg create DeptPR -rdf_consistency -type ANY
symcg -cg DeptPR addall dev -sel_rdfg 20 -sid 197300076
```

2. Set the name of RDF group 20 to SiteB and the name of RDF group 21 to SiteC:

```
symcg -cg DeptPR set -name SiteB -rdfg 000197300076:20
symcg -cg DeptPR set -name SiteC -rdfg 000197300076:21
```

3. Add devices local devices B8:BF as TGT devices:

```
symcg -cg DeptPR addall dev -devs b8:bf -tgt
```

4. Add devices B8:BF in RDF group 20 as RTGTs:

```
symcg -cg DeptPR addall dev -devs b8:bf -tgt -rdf -sid 000197300076
-rdfg 20
```

5. Add devices B8:BF in RDF group 21 as RTGTs:

```
symcg -cg DeptPR addall dev -devs b8:bf -tgt -rdf -sid 000197300076
-rdfg 21
```

Figure 125 on page 365 shows the resulting configuration:

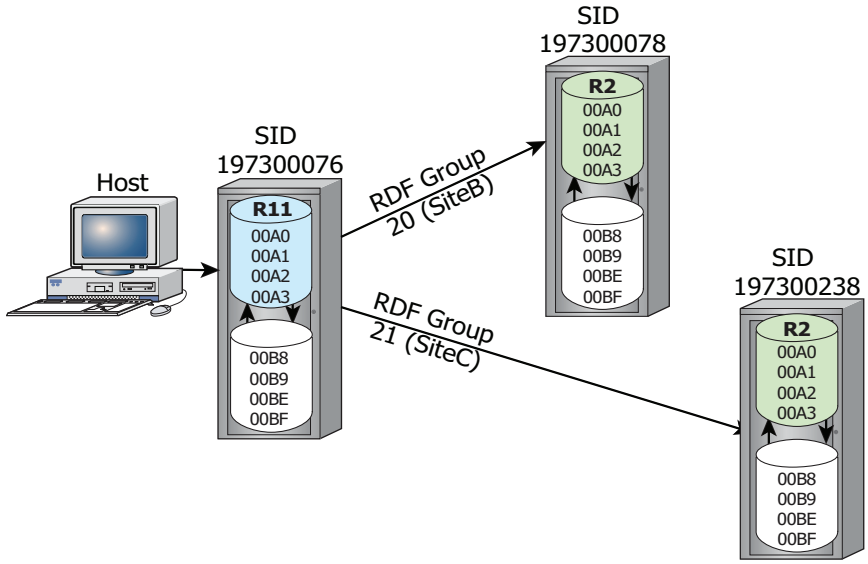


Figure 125 SnapVX and Concurrent SRDF

The following examples use the configuration shown in Figure 125 on page 365:

- Create, activate, and link a SnapVX snapshot (named LocalSnap) on the local array:

```
symsnapvx -cg DeptPR establish -name LocalSnap
symsnapvx -cg DeptPR -snapshot_name LocalSnap link
```

- Create, activate, and link a SnapVX snapshot (named SiteBSnap) of devices in RDF group SiteB at remote array 197300078:

```
symsnapvx -cg DeptPR establish -name SiteBSnap -rdfg name:SiteB -rdf
symsnapvx -cg DeptPR -snapshot_name SiteBSnap -rdfg name:SiteB -rdf
link
```

- Create, activate, and link a SnapVX snapshot (namedSiteCSnap) on devices in RDF group SiteC at the remote array 197300238:

```
symsnapvx -cg DeptPR establish -name SiteCSnap -rdfg name:SiteC -rdf
symsnapvx -cg DeptPR -snapshot_name SiteCSnap -rdfg name:SiteC -rdf
link
```

CHAPTER 12
Automated Recovery Operations

This chapter describes the following topics:

- Overview 368
- Requirements and strategies 368
- symrecover command 369
- Starting SRDF Automated Recovery 370
- Stopping SRDF Automated Recovery 371
- symrecover options file parameters 372

Overview

SRDF Automated Recovery is a utility for optimizing ever-ready fault management responses in basic SRDF environments.

SRDF Automated Recovery runs in the background and monitors the state of various SRDF/S or SRDF/A sessions.

SRDF Automated Recovery is implemented by the `symrecover` command.

If SRDF Automated Recovery detects a session failure, it attempts an automatic recovery and restart of the session. The restart uses preconfigured settings specified in the `symrecover` options file.

The options file also provides parameters for:

- Setting up error logging and event notification through email, and
- Parameters for monitor, recovery strategies, and restart actions.

Note: SRDF Automated Recovery is not supported in SRDF/Star environments.

Figure 126 shows a basic SRDF recover environment:

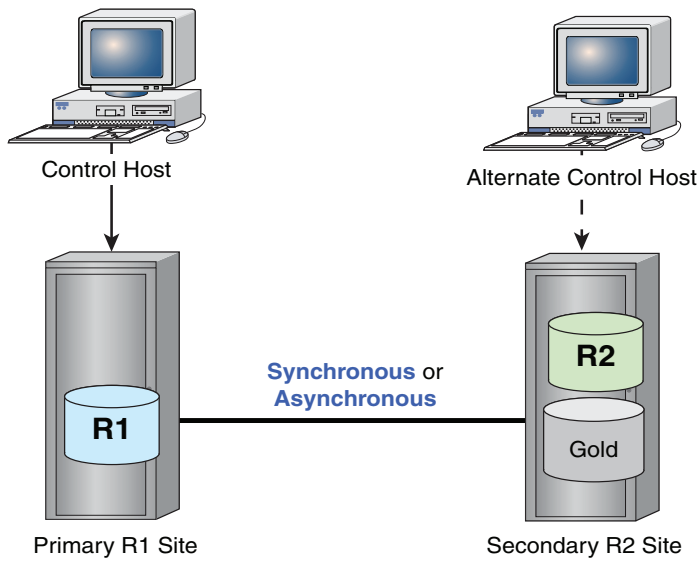


Figure 126 SRDF Recovery environment

In a basic recovery environment, a primary R1 site replicates to the secondary R2 site over a synchronous or asynchronous link.

A gold copy (BCV or clone) can be built on the R2 site to augment recovery restart strategies.

Requirements and strategies

- The `symrecover` session must be started either at the primary R1 site or the remote R2 site.
- Solutions Enabler binaries must either be in the PATH or specified as a parameter.

- The `symrecover` command can only be run with the Perl script shipped with Solutions Enabler.
- If the group is concurrent, then `symrecover` must be run from the R1 workload site.
- The initial group state must be CONSISTENT or SYNCHRONIZED, depending on the target SRDF state, unless the `restart_group_on_startup` option is specified (not the default).

Consistency protection restrictions

- If consistency protection is desired, it must be enabled prior to starting `symrecover`. A `symrecover` session must be started on the same site where consistency was enabled via a consistency group.

  Note: If you are managing using device groups, `symrecover` can be started at other sites.

Gold copy restrictions

- You can perform R2 gold copying with either Native Clones or Business Continuance Volumes (BCVs).
- BCV-to-STD association for the R2 gold copy is dynamic using the `symmir` defaults.

Restart restrictions

Note: See Table 42 on page 372 for a complete list of parameters and optional recovery actions to be set in the `symrecover` options file.

- A recovery fails if monitoring a leg that has an R22 device when the other SRDF mirror of the R22 is read/write (RW) on the link (such states as `synchronized`, `syncinprog`, or `consistent`).
- The recovery does not start when the `-restart_group_on_starup` parameters are specified, and an R22 device has another SRDF mirror that is already RW on the link.

symrecover command

SRDF Automated Recovery is launched and optimized by the `symrecover` command.

symrecover command syntax

```
symrecover [-h]
symrecover [-env | -version]

symrecover start {-g DgName | -cg CgName}
[-mode {SYNC | ASYNC}] [-out LogPath]
[-options OptnFile]
```

Where:

- Either a device group (`-g DgName`) or composite group (`-cg CgName`) must be specified.
- `-mode` - specifies the type of SRDF session. Valid values are SYNC and ASYNC. There is no default and this option must be specified.

- `-out LogPath` - specifies an alternate fully-qualified directory location to use for the log file.
- `-options OptnFile` - specifies the fully-qualified file containing program options.

Usage notes

- The symrecover command does not support the monitoring or recovery of a device group or composite group that is set with an ANY group type.
- Any options specified on the command line take precedence over the options specified by `-options OptnFile`.
- In a cascaded SRDF environment:
  - Specify the target composite group.
  - Do not use the `-mode` option.

Note: The `symrecover` command returns an error if used with an SRDF device pair containing thin and standard devices. The thin device must be on a array running Enginuity 5875 or higher. The standard device must be on a array running Enginuity 5671, 5773.50154, or 5875 and higher (but not Enginuity 5874).

Examples To start a recovery in a basic SRDF/S environment:

```
symrecover start -g DgName -mode sync -options OptnFile
```

To start a recovery in a cascaded SRDF environment:

```
symrecover start -cg CgName -cascaded_monitor_both_hops -options
OptnFile
```

Starting SRDF Automated Recovery

The `symrecover` command can be invoked manually from the command line, but more commonly is configured to run continuously in the background using:

- Windows Scheduled Tasks,
- UNIX CRON/scheduled task, or
- UNIX (RC.2) file.

You can run the `symrecover` command from either the R1 or the R2 side as long as all the SRDF standard devices in the device group or the composite group are local to the host.

When devices in groups are not local to a host, they are marked as invalid, stop all control operations from being performed against them.

If an SRDF/A group becomes synchronous (SRDF/S), `symrecover` attempts to reset the SRDF link to SRDF/A mode.

You can define devices in groups on the R2 side with a corresponding partner but `symrecover` cannot start in this environment because you cannot monitor groups on the R2 side when the remote partner is concurrent. You must monitor these groups from the host.

Examples To manually start `symrecover` for an SRDF/A composite group named `RDFAmon`, using the options file named `cg_mon_opts`:

```
symrecover start -cg RDFAmon -mode async -options cg_mon_opts
```

where the `cg_mon_opts` options file includes the following settings and default values for a BCV gold copy:

```
# Options file for symrecover
#######################################################
goldcopy_clone_list = TGT
goldcopy_location = R2
goldcopy_max_wait = 1800
goldcopy_resync_interval = 0
goldcopy_state_post_restart = ACTIVATED
goldcopy_state_startup = ACTIVATED
goldcopy_type = CLONE
help = 0
log_level = 3
monitor_cycle_time = 300
monitor_only = 0
out = /var/symapi/log
restart_adcopy_resynch_threshold = 30000
restart_attempt_pause = 60
restart_delay = 30
restart_group_on_startup = 0
restart_max_attempts = 5
restart_max_wait_adcopy_sync = 0
restart_max_wait_state_change = 0
restart_max_wait_warn_interval = 600
restart_rdfa_min_cycle_warn_interval = 300
restart_rdfa_min_cycle_warn_value = 0
restart_state_syncinprog_wait_time = 120
restart_state_syncinprog_warn_interval = 300
restart_state_transmit_wait_time = 120
restart_state_transmit_warn_interval = 300
restart_sync_type = ADCOPY
restart_window = 3600
run_once = 0
run_until_first_failure = 0
```

To recover a cascaded SRDF environment, add the following parameter settings to the options file in the previous example:

```
cascaded_monitor_both_hops = 1
goldcopy_location = All
```

These options:

- Allow recovery on both hops, and
- Build gold copies at the R21 and R2 sites.

  The hop2 (R21->R2 link) restarts quickly and safely in ADCOPY mode, during the R2 resynchronization period.

Stopping SRDF Automated Recovery

To stop `symrecover` manually, enter a Ctrl/C.

To stop a `symrecover` task running in the background use one of the following options:

- Windows - Cancel the task in the Scheduled Tasks, or use End Task in the Task Manager.
- UNIX - Use the kill command.

symrecover options file parameters

Table 42 describes the valid settings in the `symrecover` options file.

Many of these settings in Table 42 have Boolean values of 0 or 1.

- 0 disables the setting
- 1 enables the setting

Table 42 symrecover options file parameters (page 1 of 5)

Setting	Description
`cascaded_monitor_both_hops= [0|1]`	0 - (default) Monitors/recovers a single hop only invoked from any site. 1 - For cascaded SRDF environments, the `symrecover` session monitors both hops linking the cascaded sites. The `symrecover` session ignores the `-mode` option and can be invoked at either the R1 primary or the remote R2 tertiary site only (not at R21 site).
`email_addr_target= <e_addr1, e_addr2, ..., ...>`	Email notification address on errors. If any of the `email_*` options are specified, then this option must also be specified to activate email alerts. Multiple comma delimited addresses may be specified. There is no default value.
`email_addr_source= e_addr1`	Specifies an address that will be used as the 'from' field of any e-mails that symrecover sends. No checks are done about the validity of the e-mail address. If this is not specified, then a default value is generated based on the array's hostname and current user account.
`email_server= e_srvr_addr`	Specifies the host target email server. If any of the `email_*` options are specified then this option must also be specified to activate email alerts. There is no default value.
`email_subject= err_subject_string`	Specifies the email notification subject on errors. The default value is: `SymRecover Alert: Host [HostName] Group [GrpName]`
`email_log_level= SeverityLevel`	The severity level desired for the email alert triggering message. Valid values are: 0 = Off. 1 = Only Errors are reported. 2 = Errors and Warnings are reported. 3 = Errors, Warnings, and Informational messages are reported. 4 = All messages are reported, including all SYMCLI commands and responses. **Note:** For each message that meets the particular logging level requirement, an email is sent with that message. It is highly recommended that at most this be set to either a 1 or a 2. If the required email options (`email_server` and `email_addr_target`) are not specified, then the default value is 0. If they are specified then the default value is 1.

Table 42 symrecover options file parameters (page 2 of 5)

Setting	Description
`goldcopy_location= LocationValue`	Specifies the location of the backup gold copy. Possible (case insensitive) values are: • NONE = No gold copy is desired. All other gold copy optional parameters in this list are ignored. • R2 = A gold copy on the R2 site is desired. R2 is the default setting. Any R2 BCV pairs need to be already defined before calling `symrecover`.

`goldcopy_type=CopyType`
 	
Old alternate, if still necessary:
`goldcopy_type_r2= CopyType`
	Specifies the type of goldcopy to create on the R2 side. Valid (case insensitive) values are:
	none = No gold copy is desired. All other `goldcopy_*` options are ignored.
	bcv = BCV gold copy on the R2 side is created, which is the default.
	clone = Clone gold copy on the R2 is created.
	
	Note: For the BCV gold copy, the R2 BCVs must be paired with the R2 devices before starting `symrecover`.
	
	For the clone gold copy, the target devices must have a clone session with the R2 devices before starting `symrecover`.

`goldcopy_state_startup= CopyType`
 	
Old alternate, if still necessary:
`goldcopy_bcv_r2_mirror_state_startup= CopyState`
	Specifies the desired state of the R2 gold copy upon routine startup. Valid (case insensitive) values are:
	establish = the devices must be established (BCV gold copy only).
	split = The devices must be split (BCV gold copy only).
	activated = The devices must be in the copied state (clone gold copy only).
	created = The devices must be in the precopy state (clone gold copy only).
	none = The devices must be unchanged, which is the default.
	
	Note: If the gold copy type is BCV and the default state of the BCVs is establish, this has been shown to increase SRDF/A session drops.

`goldcopy_state_post_restart= CopyState`
 	
Old alternate, if still necessary:
`goldcopy_bcv_r2_mir_state_post_restart = CopyState`
	Following a successful SRDF/A session restart or BCV resync, specifies which state the R2 gold copy should be. Valid (case insensitive) values are:
	establish = The devices must be left established (BCV gold copy only).
	split = The devices must be split, which is the default (BCV gold copy only).
	activated = The devices must be in the copied state (clone only).
	created = The devices must be in the precopy state (clone only).
	
	Note: If the gold copy type is BCV and the default state of the BCVs is establish, this has been shown to increase SRDF/A session drops.

`goldcopy_max_wait= MaxWaitTime`
 	
Old alternate, if still necessary:
`goldcopy_max_wait_bcv= MaxWaitTime`
	Specifies the length of time, in seconds, for `symrecover` to wait for synchronization. Valid values are 0 to `maxint` (2147483647).
	The default is 0, which indicates for `symrecover` to wait forever.
	For clone gold copies, if the `goldcopy_state_post_restart` option is set to `activated`, it waits for the clone copied state to be reached before performing synchronization. If this option is set to `created`, it waits for the clone precopied state to be reached.

Table 42 symrecover options file parameters (page 3 of 5)

Setting	Description

`goldcopy_resync_interval= resynctime`
 	
Old alternate, if still necessary:
`goldcopy_bcv_r2_mirror_resync_interval = resynctime`
	Defines the resync interval, in minutes, for `symrecover` to automatically create a new clone gold copy or a new BCV gold copy, which overrides the existing gold copy. This action only takes place during non-error periods.
	Valid values are 0, and 15 to `maxint`. Zero (0) indicates that the mirrors are never to be automatically synchronized outside of error-producing events.The default is 15.
	Note: If the gold copy type is `BCV`, the act of frequently synchronizing the R2 BCVs has been shown to increase SRDF/A session drops.

`goldcopy_clone_list= List`	For a clone gold copy, this option tells `symrecover` which list within the device group or the composite group to search for clone devices. Valid (case insensitive) values are: tgt = Uses the TGT list. bcv = Uses the BCV list.
`monitor_cycle_time= cycletime`	Defines the number of seconds to pause between monitor status scans. The minimum value is 30 seconds, the maximum is 3600 seconds. The default value is 300 seconds.
`monitor_only= [0|1]`	Specifies to only monitor the state of specified group. No recovery actions will take place. This option is not enabled by default. **Note:** `monitor_only`, `run_once`, and `run_until_first_failure` are mutually exclusive options.
`run_once= [0|1]`	Specifies to check the status of the group once. If the group needs recovery actions perform them. Exit after one check. This option is not enabled by default. This option ignores the setting of `restart_max_attempts`. **Note:** `monitor_only`, `run_once`, and `run_until_first_failure` are mutually exclusive options.
`run_until_first_failure= [0|1]`	Specifies to monitor the group until the first failure occurs and then exit without performing any recovery action. This option is not enabled by default. This option ignores the setting of `restart_max_attempts`. **Note:** `monitor_only`, `run_once`, and `run_until_first_failure` are mutually exclusive options.
`rdfg= rdfgvalue`	When working with device groups or composite groups that contain concurrent devices, symrecover supports monitoring only one of the SRDF groups that contain mirrors of the concurrent devices. Use the rdfg option to indicate the SRDF group that symrecover should monitor. Note that monitoring of concurrent SRDF defined groups is only supported when symrecover is executed from the R1 side. The value is taken directly as specified and no data validation is performed on it. This option is not set by default and non-concurrent SRDF groups are assumed. **Note:** If the group is a composite group, and consistency is enabled, this must be of the "name:" format and this value is case sensitive.
`restart_adcopy_resynch_threshold= tracks`	Specifies the number of tracks outstanding that during recovery will trigger a switch over to SRDF/A or SRDF/S. The default value is 30000.

Table 42 symrecover options file parameters (page 4 of 5)

Setting	Description
`restart_attempt_pause= time`	Inserts a specified wait time before an attempt is made to restart a failed session to allow for things to settle down. After the `restart_attempt_pause` is complete, `symrecover` redrives the overall monitor loop. If there is still a problem, the restart failure count is incremented and a restart is attempted. Valid values are 30 to 3600 seconds. The default is 60 seconds.
`restart_delay= time`	Inserts a specified wait time after an attempt is made to restart a failed session and the attempt itself fails. Valid values are 0 (no delay, immediately restart) to maxint. The default is 30 seconds.
`restart_group_on_startup= [0|1]`	On `symrecover` startup, if the group being monitored is not initially in a Consistent state (for SRDF/A) or a Synchronized state (for SRDF/S), `symrecover` considers this an error condition and exits. If this option is specified, `symrecover` will attempt to recover the group on startup. This option is not enabled by default.
`restart_max_attempts= attempts`	Specifies the maximum number of restart attempts that are performed within the `restart_window` interval. After this limit is reached the program will terminate. The range is from 0 to maxint. The value of 0 means to attempt indefinitely.The default is 5 attempts.
`restart_max_wait_adcopy_sync= time`	Specifies the length of time (in seconds) during a restart for a program to wait for a group to achieve the `restart_adcopy_resync_threshold` number of tracks pending. Valid values are 0 to maxint. The value of 0 means to infinitely wait. The default is 0.
`restart_max_wait_state_change= statetime`	Specifies the length of time (in seconds) during a restart for a program to wait for a group to change to a desired state (once requested). Valid values are 0 to maxint. The value of 0 means to infinitely wait. The default is 0.
`restart_max_wait_warn_interval= warntime`	Specifies the length of time (in seconds) during a restart, while waiting for a state change to occur, to display a progress warning message. Valid values are 0 and 30 to maxint. The value of 0 means to wait forever. The default is 600 seconds.
`restart_rdfa_min_cycle_warn_interval= cyclewarntime`	Specifies the length of time (in seconds) before repetitively displaying a warning when the RDFA minimum cycle time exceeds the `restart_rdfa_min_cycle_warn_value` parameter. Valid values are 30 to `maxint`.The default is 600.
`restart_rdfa_min_cycle_warn_value= warntime`	Specifies the maximum value (in seconds) to which a trigger can occur with a warning message, indicating the RDFA minimum cycle time has exceeded this value. Valid values are 0 and 30 to `maxint`. The value of 0 means this feature is turned off, which is the default.
`restart_state_syncinprog_wait_time time`	The maximum length of time (in seconds) during a group syncinprog state that sleep is done before rechecking the group status. Valid values are [30] to [maxint]. The default is [120] seconds.
`restart_state_transmit_warn_interval= time`	Specifies the interval of time (in seconds) that while a group remains in a transmit idle state, to generate a warning message. Valid values are 0 to `maxint`. The default is 300 seconds.

Table 42 symrecover options file parameters (page 5 of 5)

Setting	Description
`restart_state_transmit_wait_time= transwaittime`	Specifies the maximum length of time (in seconds) that during a group transmit idle state, a sleep is done before rechecking the group status. Valid values are 30 to maxint. The default is 120 seconds.
`restart_sync_type= synctype`	Specifies the type of synchronization to be used following the detection of a failed SRDF/A session. Valid values are: ADCOPY = adaptive copy disk (default). SYNC = synchronous mode. NONE = No intermediate track resynch stage will be attempted. A direct re-establish using the existing SRDF session mode will be attempted. Note that if `cascaded_monitor_both_hops` is set, `restart_sync_type` is ignored as ADCOPY is used in the R21->R2 link at restart.
`restart_window= time`	Specifies a time window (in seconds) during which no more than restart_max_attempts failures and accompanying restart attempts will be tolerated before monitoring is terminated. The window begins at the time of the first failure and ends restart_window seconds later. A new window begins with a failure after expiration of the previous window.
`log_level= level`	The desired logging level. Valid values are: 0 = Off. 1 = Only Errors are reported. 2 = Errors and Warnings are reported. 3 = Errors, Warnings, and Informational messages are reported (default). 4 = All messages are reported.

APPENDIX A
SRDF operations and pair states

This appendix describes the following topics:

- SRDF operations and applicable pair states 378
  - "Control operations for R1 -› R2 pair states"
- Cascaded SRDF control operations and applicable pair states 383
  - "Cascaded SRDF: R1 -› R21 control operations allowed for R21-› R2 pair states"
  - "Cascaded SRDF: R21 -› R2 control operations allowed for R1 -› R21 pair states"
- Cascaded SRDF set operations and applicable pair states 389
  - "Cascaded SRDF: R1 -› R21 set operations allowed for R21 -› R2 pair states"
  - "Cascaded SRDF: R21 -› R2 set operations allowed for R1 -› R21 pair states"
- Concurrent SRDF control operations and applicable pair states 390
  - "Concurrent SRDF: control operations 1st leg R1-› 2nd leg R1 pair states"
  - "Concurrent SRDF: control operations 1st leg R2 -› 2nd leg R2 pair states"
- Consistency group control operations and applicable pair states..... 396

SRDF operations and applicable pair states

When you issue a command to perform an SRDF control operations, SRDF verifies the state of the device pairs.

If the device pair is not in a legal SRDF state to initiate the control operation, the action is blocked.

Use the `-force` option to perform the control operation, regardless of the pair state.

Note: If devices are running in SRDF/A mode, control operations for restore, update R1, and failback require the use of the -force option.

Examples:

To initiate a failover on all SRDF pairs in the `prod` group that are in the Split state:

```
symrdf -g prod -force failover
```

To initiate a failover on one SRDF pair, DEV001, in the `prod` group that is n the SyncInProg state:

```
symrdf -g prod -force failover DEV001
```

IMPORTANT

The `-force` option may place the SRDF pair into an undesirable state. After using this option, always check the pair state.

Control operations for R1 -> R2 pair states

The first column in Table 43 lists the control operations you can invoke for the listed pair states.

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.9

Table 43 SRDF control operations and applicable pair states (page 1 of 3)

Control operation:	R1 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
deletepair			Y[28,36,37]	Y[28,36,37]	Y[28,36,37]							
half_deletepair			Y[28,36,37]	Y[28,36,37]	Y[28,36]	Y[28,36,37]	Y[28,36,37]					
movepair			Y	Y	Y							

Table 43 SRDF control operations and applicable pair states (page 2 of 3)

Control operation:	R1 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
half_movepair			Y	Y	Y	Y	Y					
swap			Y[43]	Y[6,43]	Y[43]			Y[43]				
swap -refresh R1			Y[29,43]	Y[6,29,43]	Y[29,43]			Y[29,43]				
swap -refresh R2			Y[29,43]	Y[6,29,43]								
half_swap			Y[43]	Y[43]	Y[43]	Y[43]	Y[43]					
establish			Y[26]	Y[26]	Y[5,26]			Y[5,26]		Y[12]		
establish -full			Y[26]	Y[26]	Y[5,26]			Y[5,26]		Y[12, 17]		
split	Y[15,19,28,36,37]	Y		Y[19,28,36,37]				Y[19,28,36,37]			Y[19,28,36,37]	
restore			Y[29]	Y[3,29]	Y[5,29]			Y[5,29]		Y[13]		
restore -full			Y[29]	Y[3,29]	Y[5,29]			Y[5,29]		Y[13,14]		
update				Y[1,28,29, 45]	Y[28,29,45]			Y[28,29,45]				
failback			Y[26,28,29]	Y[1,26,28,29]	Y[26,28,29]	Y[8,18,26]	Y[8,18,26]	Y[26,28,29]	Y[26,28,29]			
failover	Y[15,19,30,41,42]	Y[19,30,41,42]	Y[18,19,30,41,42]	Y[19,30,41,42]		Y[9,19,41,42]	Y[9,19,41,42]	Y[19,30,41,42]	Y[15,19,30,41,42]	Y[18,19 30, 41,42]	Y[19,30,41,42]	Y[19,30,41,42]
failover -establish	Y[15,17,29,30 43]	Y [29,30, 43]	Y[18,19,30,43]	Y[29,30,43]	Y[29,30,43]			Y[29,30,43]	Y[15,29,30,43]	Y[18,29,30,43]		
failover -restore	Y[15,26,31, 43]	Y[26,31,43]	Y[18,26,31,43]	Y[26,31,43]	Y[18,26,31, 43]					Y[18,26,31,43]	Y [26,31 43]	
invalidate R1				Y[2,29]								
invalidate R2				Y[26]								
merge				Y[3,33,34]	Y[18,33,34]							
msc_cleanup			Y	Y	Y	Y[9]	Y[9]					
not_ready R1	Y[6,21]	Y[6,21]	Y[6,21]	Y[6,21]	Y[21]	Y[6,8,21]	Y[6,8,21]	Y[21]	Y[21]	Y[6,21]	Y[6,21]	Y[6,8,21]
not_ready R2	Y[10,17,21]	Y[10,21]	Y[10,21]	Y[21]	Y[10,21]	Y[7,9,21]	Y[7,9,21]	Y[10,21]	Y[10,21]	Y[10,17,21]		
ready R1	Y[21]	Y[21]	Y[21]	Y[21]	Y[21]	Y[8,21]	Y[8,21]	Y[21]	Y[21]	Y[21]	Y[21]	Y[21]

Table 43 SRDF control operations and applicable pair states (page 3 of 3)

Control operation:	R1 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
ready R2	Y[17,21]	Y[21]	Y[21]	Y[21]	Y[21]	Y[9,21]	Y[9,21]	Y[21]	Y[21]	Y[17,21]		
refresh R1				Y[1,20,39]	Y[18,39]							
refresh R2				Y[1,18,26]								
resume				Y								
suspend	Y[15,19,28,36,37,38]	Y	Y[16,18,19,28,36,37,38]	Y[7,19,28,36,37,38]	Y[5,19,28,36,37,38]	Y[19,28,36,37,38]	Y[19,28,36,37,38]	Y[19,28,36,37,38]	Y[15,19,28,36,37,38]	Y[18,19,28,36,37,38]	Y[19,20,28,36,37,38]	Y[19,20,28,36,37,38]
disable	Y[41]	Y[41]	Y[41]	Y[41]	Y[41]	Y[18,41]	Y[18,41]	Y[41]	Y[41]	Y[41]	Y[41]	Y[23]
enable	Y[24]		Y[24]	Y[24]	Y[24]			Y[24]	Y[24]	Y[24]	Y	Y[23]
rw_disable R2	Y[17]	Y	Y[11]	Y	Y[11]	Y[7,9]	Y[7,9]	Y[11]	Y[11]	Y[2,17]		
rw_enable R1	Y[1]	Y[1]		Y[1]	Y[18]	Y[1,8]	Y[1,8]			Y[1]	Y[1]	Y[1,8]
rw_enable R2				Y[7]		Y[7,9]	Y[7,9]					
write_disable R1	Y[18]	Y[18]	Y[18]	Y[18]		Y[4,8]	Y[4,8]			Y[18]	Y[18]	Y[4,8]
write_disable R2			Y[18]	Y[18]	Y[18]	Y[9,10]	Y[9,10]	Y[18]	Y[18]	Y[18]		Y[9,10]
activate -rdfa_dse	Y[22]										Y	Y[23]
deactivate -rdfa_dse	Y[22]										Y	Y[23]
activate -rdfa_devpace	Y[22,44]										Y[44]	Y[23,44]
deactivate -rdfa_devpace	Y[22,44]										Y[44]	Y[23,25,44]
activate -rdfa_pace	Y[22]										Y	Y[23]
deactivate -rdfa_pace	Y[22]										Y	Y[23,25]
activate -rdfa_wpace	Y[22]										Y	Y[23]
deactivate -rdfa_wpace	Y[22]										Y	Y[23]
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y			Y	Y		Y	
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y			Y	Y		Y	

1. SA is Write Disabled or is Not Ready on the source side.
2. SA is Write Disabled, or is Not Ready on the source side, or must use -nowd.
3. SA is Write Disabled, or is Not Ready on the source side, or must use -force.
4. SA is Ready on the source side.
5. Source is not visible to any host.
6. Write Disabled on the source.
7. SA or RA is Write Disabled or is Not Ready on the target side.
8. Host application running while connected to the source.
9. Host application running while connected to the target.
10. RA is Ready on the target side.
11. RA is Write Disabled on the target side.
12. Source and target are Not Ready but the SRDF link is Ready and there are no local or remote invalid tracks on the source or the target.
13. Source and target are Not Ready but the SRDF link is Ready and there are no remote invalid tracks on the source side.
14. Source and target are Not Ready but the SRDF link is Ready and there are no local or remote invalid tracks on the source side.
15. Can use -symforce.
16. Write Disabled on the SRDF link.
17. Not allowed when SRDF/A is active.
18. Must use -force.
19. If enabled for SRDF consistency protection, must use -force.
20. Must use -immediate.
21. Not allowed on a diskless device.
22. SRDF/A must be active.
23. Source must be reachable.
24. Must be in async mode.
25. Only allowed on the R1 side and must use -symforce.
26. No local invalid tracks on the source side.
27. No remote invalid tracks on the source side.
28. If remote invalid tracks are on the source side, must use -force.
29. No local invalid tracks on the target side.
30. If remote invalid tracks are on the source side, must use -symforce.
31. If remote invalid tracks are on the target side, must use -force.
32. No remote invalid tracks on the target side.
33. Source device is Read Write Enabled and there are no local and remote invalid tracks on the target side.
34. Target device is Read Write Enabled and there are no local and remote invalid tracks on the target side.
35. Source and target devices are Read Write Enabled.
36. If there are local invalid tracks on the source side, must use -symforce if the source is not an R11 or R21.
37. If there are local invalid tracks on the target side, must use -symforce if the target is not an R11 or R21.

38. Write Disabled on the SRDF link and must use -force.
39. No local invalid tracks on the target side and no remote invalid tracks on the source side and must use -force.
40. If the SRDF device pair is operating in adaptive copy mode, must use -symforce.
41. If enabled for CG SRDF consistency protection, must use -force.
42. If local invalid tracks are on the target side, must use –symforce.
43. Not allowed if enabled for SRDF consistency protection.
44. Not allowed if the R1 or R2 array is running Enginuity 5977 or higher
45. Not allowed if the R1 array is running Enginuity 5876 and the R2 array is running Enginuity 5977 or higher.

Cascaded SRDF control operations and applicable pair states

Allowable control operations vary depending on the type of SRDF device. This section describes allowable operations by device pair types in cascaded configurations.

Cascaded SRDF: R1 -> R21 control operations allowed for R21-> R2 pair states

Table 44 lists the allowable control operations for the R1->R21 pair given the pair states for the R21->R2 pair.

Allowed actions are noted by Ys.

Note: The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered.
The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 44 R1->R21 cascaded SRDF control operations and applicable pair states (page 1 of 3)

R1 ->R21 control operation:	R21 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
createpair -establish	Y[1,3,4,16,17,21,22]	Y[1,3,21,22]	Y[1,12,21,22]	Y[1,12,21,22]	Y[1,12,20,22]	Y[1,12,21,22]					Y[1,3,4,16,17,22]	Y[1,3,4,16,17,22]
createpair -restore	Y[1,5,16,17,22]	Y[1,22]	Y[1,12,21,22]	Y[1,12,21,22]	Y[1,12,20,22]	Y[1,6,12,22]					Y[1,6,16,17,22]	Y[1,5,16,17,22]
createpair -invalidate R1	Y[1,5,16,22]	Y[1,6,22]	Y[1,22]	Y[1,22]	Y[1,22]	Y[1,22]		Y[1,22]	Y[1,22]		Y[1,16,22]	Y[1,5,16,22]
createpair -invalidate R2	Y[1,6,16,22]	Y[1,6,22]	Y[1,6,22]	Y[1,6,22]	Y[1,6,22]	Y[1,6,22]		Y[1,6,22]	Y[1,6,22]		Y[1,16, 22]	Y[1,16, 22]
deletepair	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,12]	Y[1,12]	Y[1,12]	Y[1,13]	Y[1,12]	Y[1,12]
half_deletepair	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,12]	Y[1,12]	Y[1,12]	Y[1,13]	Y[1,12]	Y[1,12]
movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
swap	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]		Y[1,12]	Y[1,12]		Y[1,12]	Y[1,12]
half_swap	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,12]	Y[1,12]	Y[1,12]		Y[1,12]	Y[1,12]
swap -refresh R1	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]		Y[1,12]	Y[1,12]		Y[1,12]	Y[1,12]
swap -refresh R2			Y[1]	Y[1]	Y[1,20]	Y[1]						
establish	Y[4,17]	Y[2]	Y[12]	Y[12]	Y[12,20]	Y[12]					Y[2,4,17]	Y[2,3,4,17]
establish -full	Y[3,4,17]	Y[3]	Y[12]	Y[12]	Y[12,20]	Y[12]					Y[2,3,4,17]	Y[2,3,4,17]

Table 44 R1->R21 cascaded SRDF control operations and applicable pair states (page 2 of 3)

R1 ->R21 control operation:	R21 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y[5,17]	Y	Y[12]	Y[12]	Y[12,20]	Y[6,12]					Y[17]	Y[5,6,17]
restore -full	Y[5,17]	Y		Y[12]	Y[12,20]	Y[6,12]					Y[17]	Y[5,6,17]
update	Y[5,17]	Y	Y[12]	Y[12]	Y[12]	Y[12]		Y	Y		Y[17]	Y[5,17]
failback	Y[5,17]	Y	Y[12]	Y[12]	Y[12,20]	Y[6,12]					Y	Y[5,6,17]
failover	Y	Y	Y	Y	Y	Y	Y			Y	Y	Y
failover -establish	Y[1,12]	Y[1,12]	Y[1,12]	Y[1,12]	Y[1,12,20]	Y[1,12]					Y[1,12]	Y[1,12]
failover -restore	Y[1,8,11,12]	Y[1,11,12]	Y[1,12]	Y[1,12]	Y[1,12,20]	Y[1,12]	Y[1,11,12]			Y[11,12]	Y[1,8,11,12]	Y[1,8,11,12]
invalidate -R1	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
merge	Y[14]	Y[14]	Y[14]	Y[14]	Y[14]	Y[14]		Y[14]	Y[14]		Y[14]	Y[14]
msc_cleanup												
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
refresh R2			Y	Y	Y	Y		Y	Y			
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y[4,5,17]	Y[5]	Y[12]	Y[12]	Y[12,20]	Y[12]					Y[2,4,17]	Y[2,3,4,5,17]
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 44 R1->R21 cascaded SRDF control operations and applicable pair states (page 3 of 3)

R1 ->R21 control operation:	R21 ->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. If the other pair (the one not being controlled) is enabled for SRDF consistency protection, must use -force. This operation can change the composite group type, causing SRDF consistency monitoring to stop.
2. If the other pair (not being controlled) is enabled for SRDF consistency protection, must use -force.
3. Must use -force.
4. Not allowed if operation results in R1->R21<-R2 data resychronization.
5. Not allowed if R21 is diskless and operation will result in R1<-R21->R2 data resynchronization.
6. If tracks are owed to R21 while R21->R2 is in the Transmit Idle state, data resynchronization between R1->R21 cannot complete.
7. Not allowed if operation results in local invalid tracks on the R21 device.
8. Not allowed if R2 owes tracks to R21.
9. Not allowed if operation results in data flowing from R2->R21.
10. Not allowed if operation creates a concurrent R22 device on a array running on an Enginuity level lower than 5773.150.
11. Must use -remote.
12. Not allowed if R21 is diskless.
13. Not allowed if R1 is diskless and the SRDF link of the other pair is RW.
14. Not allowed if R21 is diskless and both mirrors of R21 have invalid tracks.
15. Must use -force. The state of the other pair changes to Suspended.
16. Not allowed if SRDF/A group-level write pacing or SRDF/A device-level write pacing is active and supported on the R1 mirror of what will become the R21 and the R21 array is running an Enginuity level lower than 5876 Q42012 SR.

17. If the pair being controlled is the R1->R21 pair and is operating in adaptive copy mode and the R1 mirror of the R21 has either SRDF/A group-level or SRDF/A device-level write pacing activated and supported, must use -force.
18. If the pair being controlled is (or will become) the R21->R2 pair and is operating in asynchronous mode with SRDF/A device-level or group-level write pacing configured for autostart on the R1 mirror of the R21, and the R1->R21 pair is operating in adaptive copy mode and is read/write (RW) on the SRDF link, must use -force.
19. If the R1->R21 pair is operating in adaptive copy mode and is read/write (RW) on the SRDF link, must use –force.
20. The R21 is not visible to any host.
21. Not allowed when what will become the R21->R2 is in Synchronous mode.
22. Not allowed if the R21 (or what will become the R21), R1 or R2 array is running HYPERMAX OS/Solutions Enabler 8.0.1 or higher.

Cascaded SRDF: R21 -> R2 control operations allowed for R1 -> R21 pair states

Table 45 lists the allowable control operations for the R21->R2 pair given the SRDF pair states for the R1->R21 pair.

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 45 R21->R2 cascaded SRDF control operations and applicable pair states (page 1 of 3)

	R1 ->R21 pair state:											
R21->R2 control operation:	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
createpair -establish	Y[1,18, 22]	Y[1,18, 22]	Y[1,5, 22]	Y[1,5, 22]	Y[1,5,22]	Y[1,5, 22]		Y[1, 22]	Y[1, 22]		Y[1, 22]	Y[1, 22]
createpair -restore			Y[1,6, 22]	Y[1, 22]	Y[1,6, 22]	Y[1,6, 22]		Y[1, 22]	Y[1, 22]			
createpair -invalidate R1	Y[1, 22]	Y[1, 22]	Y[1, 22]	Y[1, 22]	Y[1,6, 22]	Y[1,6, 22]		Y[1, 22]	Y[1, 22]		Y[1, 22]	Y[1, 22]
createpair -invalidate R2	Y[1, 22]	Y[1, 22]	Y[1,5, 22]	Y[1,5, 22]	Y[1,5, 22]	Y[1,5, 22]		Y[1, 22]	Y[1, 22]		Y[1,6, 22]	Y[1, 22]
deletepair	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,12]	Y[1,12]	Y[1,12]	Y[1]	Y[1,12]	Y[1,12]
half_deletepair	Y[1,12]	Y[1,12]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,12]	Y[1,12]	Y[1,12]	Y[1]	Y[1,12]	Y[1,12]
movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
swap	Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,2, 10, 22]		Y[1,10, 12, 22]	Y[1,2, 10, 22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]

Table 45 R21->R2 cascaded SRDF control operations and applicable pair states (page 2 of 3)

R21->R2 control operation:	R1 ->R21 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
half_swap	Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,10 22]	Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 12, 22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]
swap -refresh R1	Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1, 10,22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]
swap -refresh R2	Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1,10, 22]	Y[1, 10,22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]		Y[1,10, 12, 22]	Y[1,10, 12, 22]
establish	Y[18]	Y[18]	Y[5]	Y[5]	Y[5]	Y[5]		Y	Y		Y	Y
full establish	Y[18]	Y[18]	Y[5]	Y[5]	Y[5]	Y[5]		Y	Y		Y	
split	Y[12]	Y[12]	Y	Y	Y	Y	Y[12]	Y[12]	Y[12]	Y[13]	Y	Y
restore			Y	Y	Y	Y		Y	Y			
full restore			Y	Y	Y	Y		Y	Y			
update			Y	Y	Y	Y		Y	Y			
failback			Y	Y	Y	Y[12]		Y	Y			
failover			Y	Y	Y	Y	Y	Y[12]	Y[12]	Y[13]		
failover -establish			Y[1,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 12,20, 22]	Y[1, 10,22]						
failover -restore			Y[1,5,10, 12, 22]	Y[1,10, 12, 22]	Y[1,10, 12,20, 22]							
invalidate -R1	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
merge	Y[5,7]	Y[5,7]	Y[5]	Y[5]	Y[5,7]	Y[5,7]		Y[5,7]	Y[5,7]		Y[5,7]	Y[5,7]
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1			Y	Y	Y[20]	Y						
refresh R2	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
suspend	Y[12]	Y[12]	Y	Y	Y	Y	Y[12]	Y[12]	Y[12]	Y[13]	Y	Y
resume	Y[9,18]	Y[9,18]	Y[5]	Y[5]	Y[5]	Y[5]		Y	Y		Y[9]	Y[6,9]
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 45 R21->R2 cascaded SRDF control operations and applicable pair states (page 3 of 3)

R21->R2 control operation:	R1 ->R21 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y[19]	Y[19]	Y	Y	Y	Y		Y[19]	Y[19]		Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y[19]	Y[19]	Y	Y	Y	Y		Y[19]	Y[19]		Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y[19]	Y[19]	Y	Y	Y	Y		Y[19]	Y[19]		Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: Table 45 uses the same keys as Table 44 on page 383.

Cascaded SRDF set operations and applicable pair states

Allowable set operations vary depending on the type of SRDF device. This section describes allowable operations by device pair types in cascaded configurations.

Cascaded SRDF: R1 -> R21 set operations allowed for R21 -> R2 pair states

Table 46 lists the allowable set operations for the R1->R21 pair given the pair states for the R21->R2 pair.

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 46 R1->R21 cascaded SRDF set operations and applicable pair states

R1->R21 set operation:	R21->R2 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
set mode async[2]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y[1]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[1]	Y[1]
set mode acp_wp	Y[1]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[1]	Y[1]

1. You must use -force if SRDF/A device-level and/or group-level write pacing is activated and supported for the SRDF/A session that includes the R21->R2 RDF device pair, and the R1->R21 SRDF device pair (that is being controlled) is read/write (RW) on the SRDF link.
2. Async mode is not supported on both sides of an R21. It is only supported on one side or the other.

Cascaded SRDF: R21 -> R2 set operations allowed for R1 -> R21 pair states

Table 47 lists the allowable set operations for the R21->R2 pair given the pair states for the R1->R21 pair.

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 47 R21->R2 Cascaded RDF Set Operations and Applicable Pair States

R21->R2 set operation:	R1->R21 pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
set mode async	Y[1]	Y[1]	Y	Y	Y	Y	Y[2]	Y[1]	Y[1]	Y[2]	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Must use -force if all of these conditions apply:
   a. SRDF/A device-level and/or group-level write pacing is configured for autostart for the SRDF/A session that includes the R21->R2 SRDF device pair.
   b. The R21->R2 SRDF device pair (that is being controlled) is read/write (RW) on the SRDF link.
   c. The R1->R21 SRDF device pair (that is not being controlled) is operating in adaptive copy mode and is read/write (RW) on the SRDF link.
2. Not allowed if SRDF/A device-level and/or group-level write pacing is configured for autostart for the SRDF/A session that includes the R21->R2 device pair.

Concurrent SRDF control operations and applicable pair states

This section provides the concurrent SRDF control operations and their applicable pair states for concurrent R1 (R11) and concurrent R2 (R22).

About concurrent R1

In a concurrent R1 relationship, there are two separate links, or legs, sending data from one R1 device to two separate R2 mirrors. You can perform a control operation on one of these legs only if the other leg is in a certain pair state.

Note: If a concurrent R1 device is made RW (read write) from either of the SRDF relationships, it is also seen as RW from the other relationship. The commands to make a concurrent R1 device RW are: `rw_enable R2`, `split`, and `failover`.

Concurrent SRDF: control operations 1st leg R1-> 2nd leg R1 pair states

Table 48 lists the allowable control operations for the first leg of the concurrent R1 pair (the one being controlled by an SRDF action) given the pair state of the second leg (the one not being controlled).

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 48 SRDF control operations and applicable states for concurrent R1 pairs (page 1 of 2)

Control operation of 1st leg of concurrent SRDF R1 pair:	Pair state of 2nd leg of concurrent SRDF R1 pair:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
createpair -establish	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
createpair -restore	Y[1,5,6]	Y[1,6]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,6]			Y[1,5]	Y[1,5,6]	Y[1,5,6]
createpair -invalidate R1	Y[1,5]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,5]	Y[1]	Y[1,5]	Y[1,5]
createpair -invalidate R2	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
deletepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_deletepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
swap	Y[1,12, 17,18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17 ,18]	Y[1,17,1 8]	Y[1,17, 18]	Y[1,12, 18]	Y[1,12,18]
half_swap	Y[1,12, 17,18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17 ,18]	Y[1,17,1 8]		Y[1,12, 18]	Y[1,18]
swap -refresh R1	Y[1,5, 12,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1, 17, 18]	Y[1,5, 17, 18]	Y[1,17, 18]	Y[1,5, 12,18]	Y[1,5,12, 18]
swap -refresh R2	Y[1,12, 17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,17 18]	Y[1,17, 18]	Y[1,17, 18]	Y[1,12, 18]	Y[1,12, 18]
establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
full establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
split	Y	Y	Y	Y	Y[2]	Y	Y	Y[3]		Y	Y	Y
restore	Y[5,6]	Y[6]	Y	Y	Y	Y	Y[6]			Y[6]	Y[5,6]	Y[5,6]
full restore	Y[5,6]	Y[6]	Y	Y	Y	Y	Y[6]			Y[6]	Y[5,6]	Y[5,6]
update	Y[5,6]	Y[6]	Y	Y	Y	Y	Y			Y[6]	Y[5,6]	Y[5,6]
failback	Y[5]	Y	Y[5]	Y[5]	Y[5]	Y[5]	Y[5]	Y	Y[5]	Y[6]	Y[5]	Y[5]
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y[1,12, 13, 18]	Y[1, 18]	Y[1,8, 18]	Y[1,8, 18]	Y[1,8, 15, 18]	Y[1,8, 18]	Y[1,8, 18]				Y[1,12, 13, 18]	Y[1,12,13, 18]
failover -restore	Y[1,11, 12,13, 18]	Y[1, 18]	Y[1,8, 18]	Y[1,8, 18]	Y[1,8, 15, 18]	Y[1,8, 18]	Y[1,8, 18]				Y[1,12, 13, 18]	Y[1,4,11, 12,13, 18]
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 48 SRDF control operations and applicable states for concurrent R1 pairs (page 2 of 2)

Control operation of 1st leg of concurrent SRDF R1 pair:	Pair state of 2nd leg of concurrent SRDF R1 pair:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y[5]	Y	Y	Y	Y	Y	Y	Y	Y[5]		Y[5]	Y[5]
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y[5]	Y	Y	Y	Y	Y	Y	Y	Y[5]		Y[5]	Y[5]
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y[5,7]	Y[7]	Y	Y	Y	Y	Y[6]	Y	Y[5]	Y[5,6]	Y[5,7]	Y[5,7]
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. If the other pair (the one not being controlled) is enabled for SRDF consistency protection, must use -force. This operation can change the composite group type, causing SRDF consistency monitoring to stop.
2. Must use -force. The state of the other pair changes to Suspended.

3. Changes the state of the other pair to Split.
4. If device will become an R21 and the other pair is in Transmit Idle pair state, data synchronization between R1->R21 cannot complete.
5. Not allowed if R2->R11<-R2 data resynchronization will result.
6. Must use -remote.
7. Only allowed if data flows from R2 to R11 and -remote is used.
8. Not allowed if R11 is diskless.
9. Not allowed if a diskless device and will become an R21 device.
10. Not allowed if tracks are owed to R21 from concurrent R2.
11. Not allowed if a diskless device and will become an R21 device and results in R1<-R21->R2 data resynchronization.
12. Not allowed if SRDF/A group-level write pacing or SRDF/A device-level write pacing is active and supported on the other R1 mirror what will become the R21 and the R21 array is running an Enginuity level lower than 5876 Q42012 SR.
13. If the pair being controlled is operating in adaptive copy mode, must use -force if the R1 mirror of what will become the R21 has SRDF/A group-level and/or device-level write pacing activated and supported.
14. If the pair being controlled is operating in asynchronous mode, with SRDF/A group-level and/or device-level write pacing enabled for autostart on what will be the R1 mirror of the resulting R21, must use -force if the other pair (that is not being controlled) is operating in adaptive copy mode.
15. The R11 is not visible to any host.
16. The other pair's (the one not being controlled) R1 is not visible to any host.
17. Not allowed when what will become the R21->R2 is in Synchronous mode.
18. Not allowed if what will become the R21, R1 or R2 array is running HYPERMAX OS/Solutions Enabler 8.0.1 or higher.

Concurrent SRDF: control operations 1st leg R2 -> 2nd leg R2 pair states

Concurrent R2 devices are intended for SRDF/Star configurations.

In a concurrent R2 configuration, an R2 device has two remote mirrors, only one of which can be active (read/write) at a given time. *EMC VMAX3 Family Product Guide* provides more information.

Table 49 lists the allowable control operations for the first leg (the one being controlled by an SRDF action) of the concurrent R2 pair given the pair state of the second leg (the one not being controlled).

Allowed actions are noted by Ys.

The Partitoned1 pair state indicates that the remote array is in the SYMAPI database and was discovered. The Partitioned2 pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

Table 49 SRDF control operations and applicable states for concurrent R2 pairs (page 1 of 2)

Control operation of 1st leg of concurrent SRDF R2 pair:	Pair state of 2nd leg of concurrent SRDF R2 pair:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
createpair -establish			Y[1,2,18]	Y[1,18]	Y[1,2,18]	Y[1,18]						
createpair -restore			Y[1,2,18]	Y[1,18]	Y[1,2,18]	Y[1,18]						
createpair -invalidate R1	Y[1,18]	Y[1,18]	Y[1,2,18]	Y[1,18]	Y[1,2,18]	Y[1,18]		Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]
createpair -invalidate R2	Y[1,18]	Y[1,18]	Y[1,2,18]	Y[1,18]	Y[1,2,18]	Y[1,18]		Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]
deletepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_deletepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
half_movepair	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]
swap	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]
half_swap	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]
swap -refresh R1	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]		Y[1,18]	Y[1,18]
swap -refresh R2				Y[1,18]		Y[1,18]						
establish			Y[2]	Y	Y[2]	Y						
establish -full			Y[2]	Y	Y[2]	Y						
split			Y	Y[3]	Y	Y		Y	Y	Y		
restore			Y[2]	Y	Y[2]	Y						
restore -full			Y[2]	Y	Y[2]	Y						
update			Y	Y	Y	Y						
failback			Y[2]	Y	Y[2]	Y						
failover			Y	Y[3]	Y	Y	Y			Y		
failover -establish			Y[18]	Y[3,18]	Y[16,18]	Y[18]						
failover -restore			Y[1,18]	Y[1,3,18]	Y[1,16,18]	Y[1,18]		Y[1,14,18]	Y[1,14,18]			
invalidate -R1	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
merge			Y	Y	Y	Y						

Table 49 SRDF control operations and applicable states for concurrent R2 pairs (page 2 of 2)

Control operation of 1st leg of concurrent SRDF R2 pair:	Pair state of 2nd leg of concurrent SRDF R2 pair:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
msc_cleanup			Y	Y	Y	Y	Y					
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y[2]	Y	Y[2]	Y	Y	Y[2]	Y[2]	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y
refresh R2				Y		Y						
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume			Y[2]	Y	Y[2]	Y						
rw_disable R2	Y	Y	Y[2]	Y	Y[2]	Y		Y[2]	Y[2]		Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
rw_enable R2			Y	Y[3]	Y	Y	Y	Y	Y	Y		
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y[2]	Y	Y[2]	Y		Y[2]	Y[2]		Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: Table 49 uses the same keys as Table 48 on page 391.

Consistency group control operations and applicable pair states

This section provides the consistency group (SRDF/CG) control operations and the applicable pair states for devices within a consistency group.

Allowed actions are noted by Ys.

Table 50 SRDF control operations and applicable pair states for devices in an SRDF/CG

Control operation:	Pair state:											
	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned1	Partitioned2	R1 Updated	R1 UpdInProg	Invalid	Consistent	TransmitIdle
enable	Y	Y	Y	Y	Y			Y	Y	Y	Y	
disable	Y	Y	Y	Y	Y	Y[2]	Y[2]	Y	Y	Y	Y	Y[2]
modify -add	Y[1]	Y									Y	
modify -remove	Y[1]	Y									Y	
modify -recover	Y	Y		Y		Y[2]	Y[2]				Y	

1. There are no local invalid tracks on the source side and no remote invalid tracks on the remote side.
2. Must use -force.

APPENDIX B
SRDF operations and TimeFinder sessions

This appendix describes the following topics:

- SRDF operations during TimeFinder/Snap/VP Snap sessions 398
  - "SRDF operations when R1 is source of TimeFinder Snap/VP Snap"
  - "SRDF operations when R1 is target of TimeFinder Snap/VP Snap"
  - "SRDF operations when R2 is source of TimeFinder Snap/VP Snap"
  - "SRDF operations when R2 is target of TimeFinder Snap/VP Snap"
- SRDF operations for TimeFinder/Clone sessions 408
  - "SRDF operations when R1 is source of TimeFinder Clone"
  - "SRDF operations when R1 is target of TimeFinder Clone"
  - "SRDF operations when R2 is source of TimeFinder Clone"
  - "SRDF operations when R2 is target of TimeFinder Clone"
- SRDF operations for Extent-level TimeFinder/Clone sessions 418
  - "SRDF operations when R1 is source of Extent-level Clone"
  - "SRDF operations when R1 is target of Extent-level Clone"
  - "SRDF operations when R2 is source of Extent-level Clone"
  - "SRDF operations when R2 is target of Extent-level Clone"
- SRDF set operations for TimeFinder/Snap sessions 428
  - "SRDF set operations when R1 is source of TimeFinder/Snap"
  - "SRDF set operations when R1 is target of TimeFinder/Snap"
  - "SRDF set operations when R2 is source of TimeFinder/Snap"
  - "SRDF set operations when R2 is target of TimeFinder/Snap"
- SRDF set operations for TimeFinder/Clone sessions 431
  - "SRDF set operations when R1 is source of TimeFinder/Clone"
  - "SRDF set operations when R1 is target of TimeFinder/Clone"
  - "SRDF set operations when R2 is source of TimeFinder/Clone"
  - "SRDF set operations when R2 is target of TimeFinder/Clone"
- SRDF set operations for Extent-level TimeFinder/Clone sessions..... 434
  - "SRDF set operations when R1 is source of Extent-level Clone"
  - "SRDF set operations when R1 is target of Extent-level Clone"
  - "SRDF set operations when R2 is source of Extent-level Clone"
  - "SRDF set operations when R2 is target of Extent-level Clone"
- SRDF operations for TimeFinder Snap VX 436
  - "SRDF operations when R1 is source of Snapvx"
  - "SRDF operations when R1 is target for Snapvx"
  - "SRDF operations when R2 is source of Snapvx"
  - "SRDF operations when R2 is target for Snapvx"
- SRDF set operations for TimeFinder Snapvx sessions..... 444
  - "SRDF set operations when R1 is source of TimeFinder Snapvx"
  - "SRDF set operations when R1 is target of TimeFinder Snapvx"
  - "SRDF set operations when R2 is source of TimeFinder Snapvx"
  - "SRDF set operations when R2 is target of TimeFinder Snapvx"

SRDF operations during TimeFinder/Snap/VP Snap sessions

This section lists the allowable SRDF operations for TimeFinder/Snap and TimeFinder VP Snap copy sessions on the R1 source and target and the R2 source and target.

Note: TimeFinder/Snap and TimeFinder VP Snap are separate features. They are combined in this section because their interactions with SRDF are very similar.

Some footnotes in the tables below refer to devices that are not pace-capable. For additional details, see "Identifying devices that cannot be paced in a cascaded SRDF configuration" on page 144.

SRDF operations when R1 is source of TimeFinder Snap/VP Snap

Table 51 identifies the allowable SRDF actions when the R1 is the source of a TimeFinder/Snap or VP Snap session.

Table 51 Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y	Y	Y	Y	Y[3]	Y	Y		
createpair -restore	Y	Y	Y	Y	Y[2]	Y[3]	Y	Y		
createpair -invalidate R1	Y	Y	Y	Y	Y	Y[3]	Y	Y		
createpair -invalidate R2	Y	Y	Y	Y	Y	Y[3]	Y	Y		
createpair -format	Y									
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
half_swap	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
swap -refresh R1	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
swap -refresh R2	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
establish	Y	Y	Y	Y	Y	Y[3]	Y	Y		
establish -full	Y	Y	Y	Y	Y	Y[3]	Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y	Y	Y	Y	Y[3]	Y	Y		
restore -full	Y	Y	Y	Y	Y[2]		Y	Y		

Table 51 Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
update	Y	Y	Y	Y	Y	Y[3]	Y	Y		
failback	Y	Y	Y	Y	Y	Y[3]	Y	Y		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
failover -restore	Y	Y[1]	Y[1]	Y[1]	Y[1]			Y[1]		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y	Y	Y	Y[3]	Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y[3]	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y	Y	Y	Y	Y[3]	Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 51 Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the devices are in asynchronous mode and R1 and R2 array are running Enginuity 5876 or less and there is a TimeFinder/Snap off of the R1 and either:
   a. The SRDF pair is an R22->R1 of a concurrent R2 setup in which either:
      i. The R22 array is running an Enginuity level less than 5876 or the R1 array is running an Enginuity level less than 5875..
      ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.
   b. The SRDF pair is not an R22->R1 of a concurrent R2 setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.
2. Allowed for TimeFinder VP Snap. If not TimeFinder VP Snap, must use -force.
3. Only allowed for TimeFinder VP Snap.

SRDF operations when R1 is target of TimeFinder Snap/VP Snap

Table 52 identifies the allowable SRDF actions when R1 is the target of a TimeFinder/Snap or VP Snap session.

Table 52 Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -restore	Y						Y	Y		
createpair -invalidate R1	Y						Y	Y		
createpair -invalidate R2	Y						Y	Y		
createpair -format	Y									
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 52 Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
swap	Y							Y		
half_swap	Y							Y		
swap -refresh R1	Y							Y		
swap -refresh R2	Y							Y		
establish	Y						Y	Y		
establish-full	Y						Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y						Y	Y		
restore -full	Y						Y	Y		
update	Y						Y	Y		
failback	Y						Y	Y		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y							Y		
failover -restore	Y							Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y						Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y						Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y						Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 52 Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for TimeFinder/Snap R1 targets.

SRDF operations when R2 is source of TimeFinder Snap/VP Snap

Table 53 identifies the allowable SRDF actions when the R2 is the source of a TimeFinder/Snap or VP Snap copy session.

Table 53 Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y[2]	Y[2]	Y[2]	Y[2]			Y[2]		
createpair -restore	Y	Y[2]	Y[2]	Y[2]	Y[2]			Y[2]		
createpair -invalidate R1	Y	Y[2]	Y[2]	Y[2]	Y[2]			Y[2]		
createpair -invalidate R2	Y	Y[2]	Y[2]	Y[2]	Y[2]			Y[2]		
createpair -format	Y									
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 53 Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y[4]	Y[4]	Y[4]	Y[4]	Y	Y	Y[4]		
half_movepair	Y	Y[4]	Y[4]	Y[4]	Y[4]	Y	Y	Y[4]		
swap	Y	Y	Y	Y	Y		Y	Y		
half_swap	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R1	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R2	Y	Y	Y	Y	Y		Y	Y		
establish	Y	Y[2]	Y[2]	Y[2]	Y[2]		Y[2]	Y[2]		
establish -full	Y	Y[2]	Y[2]	Y[2]	Y[2]		Y[2]	Y[2]		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y[2]	Y[2]	Y[2]	Y[2]	Y[1]	Y[2]	Y[2]		
restore -full	Y	Y[2]	Y[2]	Y[2]	Y[2]	Y[1]	Y[2]	Y[2]		
update	Y	Y[2]	Y[2]	Y[2]	Y[2]		Y[2]	Y[2]		
failback	Y	Y[2]	Y[2]	Y[2]	Y[2]		Y[2]	Y[2]		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y	Y	Y	Y		Y	Y		
failover -restore	Y	Y	Y	Y	Y		Y	Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y	Y	Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y		Y	Y		

Table 53 Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y[2]	Y[2]	Y[2]	Y[2]		Y[2]	Y[2]		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]	Y[3]		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Only allowed if the required Enginuity levels or patches are detected. If TimeFinder VP Snap, requires Enginuity version 5876 Q42012 SR and higher.
2. Not allowed if the devices are in asynchronous mode and R1 and R2 array are running Enginuity 5876 or less and there is a TimeFinder/Snap off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.

3. If the SRDF/A session is in the Transmit Idle state, you must issue the command with -symforce from the R1 side.
4. Not allowed if the devices are moving to a group operating in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less, and there is a TimeFinder/Snap off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level lower than 5876, or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
5. If a TimeFinder VP Snap, you must use -force.

SRDF operations when R2 is target of TimeFinder Snap/VP Snap

Table 54 identifies the allowable SRDF actions when the R2 is the target of a TimeFinder/Snap or VP Snap copy session.

Table 54 Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y								Y		
createpair -restore	Y								Y		
createpair -invalidate R1	Y								Y		
createpair -invalidate R2	Y								Y		
createpair -format	Y										
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y					Y		Y	Y		
half_swap	Y					Y		Y	Y		
swap -refresh R1	Y					Y		Y	Y		
swap -refresh R2	Y					Y		Y	Y		
establish	Y					Y		Y	Y		
establish -full	Y					Y		Y	Y		

Table 54 Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y					Y		Y	Y		
restore -full	Y					Y		Y	Y		
update	Y					Y		Y	Y		
failback	Y					Y		Y	Y		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y					Y		Y	Y		
failover -restore	Y					Y		Y	Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y					Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y					Y		Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y					Y		Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 54 Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

SRDF operations for TimeFinder/Clone sessions

This section lists the allowable SRDF operations for TimeFinder /Clone copy sessions on the R1 source and target and the R2 source and target.

SRDF operations when R1 is source of TimeFinder Clone

Table 55 identifies the allowable SRDF actions when the R1 is the source of a TimeFinder/Clone copy session.

Table 55 Allowable SRDF operations when R1 is the source of a TimeFinder/Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -invalidate R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -invalidate R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
half_swap	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
swap -refresh R1	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
swap -refresh R2	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
establish -full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
restore -full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
update	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		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
failover -restore	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		

Table 55 Allowable SRDF operations when R1 is the source of a TimeFinder/Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the devices are in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the R1 and

either:

a. The SRDF pair is an R22->R1 of a concurrent R2 setup in which either:

   i. The R22 array is running an Enginuity level less than 5876 or the R1 array is running an Enginuity level less than 5875.

   ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.

b. The SRDF device pair is not an R22->R1 of a concurrent R2 setup and either:

   i. The Enginuity level is lower than 5875 on either the R1 or R2 array.

   ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.

SRDF operations when R1 is target of TimeFinder Clone

Table 56 identifies the allowable SRDF actions when the R1 is the target of a TimeFinder/Clone copy session.

Table 56 Allowable SRDF operations when R1 is the target of a TimeFinder/Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y						Y			Y		Y	Y		
createpair -restore	Y						Y			Y		Y	Y		
createpair -invalidate R1	Y						Y			Y		Y	Y		
createpair -invalidate R2	Y						Y			Y		Y	Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y												Y		
half_swap	Y												Y		
swap -refresh R1	Y												Y		
swap -refresh R2	Y												Y		
establish	Y					Y	Y			Y		Y	Y		
full establish	Y					Y	Y			Y		Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y						Y			Y		Y	Y		
restore -full	Y						Y			Y		Y	Y		

Table 56 Allowable SRDF operations when R1 is the target of a TimeFinder/Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
update	Y						Y			Y		Y	Y		
failback	Y						Y			Y		Y	Y		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y												Y		
failover -restore	Y												Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y						Y			Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y						Y			Y		Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y					Y	Y			Y		Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 56 Allowable SRDF operations when R1 is the target of a TimeFinder/Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for TimeFinder/Clone R1 targets.

SRDF operations when R2 is source of TimeFinder Clone

Table 57 identifies the allowable SRDF actions when the R2 is the source of a TimeFinder/Clone copy session.

Table 57 Allowable SRDF operations when R2 is the source of a TimeFinder/Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[1,3]	Y[3]	Y			Y[3]		
createpair -restore	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y			Y[3]		
createpair -invalidate R1	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y			Y		
createpair -invalidate R2	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y			Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y[5]	Y[5]	Y[5]	Y	Y	Y	Y[5]	Y[5]	Y	Y	Y	Y[5]		
half_movepair	Y	Y[5]	Y[5]	Y[5]	Y	Y	Y	Y[5]	Y[5]	Y	Y	Y	Y[5]		
swap	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
half_swap	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
establish	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
full establish	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3,1]	Y[3]	Y		Y	Y[3]		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[2]	Y	Y[3]		
restore -full	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[2]	Y	Y[3]		
update	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
failback	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
failover -restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 57 Allowable SRDF operations when R2 is the source of a TimeFinder/Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[3]	Y	Y[3]		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[3]	Y	Y[3]		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Only allowed if required Enginuity levels or patches are detected and not a TimeFinder VP Snap.
2. Not allowed if the devices are in async mode and there is a TimeFinder/Clone off of the R2 either:
   a. The SRDF pair will become an R21->R2 for which any of the following apply:
      i. The R21 array is running an Enginuity level lower than 5876 Q42012 SR, or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
      iii. If the R21->R2 pair will be read/write (RW) on the SRDF link, the R21 must be pace-capable.
   b. The SRDF device pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
3. Not allowed if the devices are in asynchronous mode and R1 and R2 array are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      i. ii.SRDF/A device-level write pacing is not configured for autostart on the R1 side.
4. Not allowed if the devices are moving to a group operating in async mode and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup and any of the following apply:
      i. The R21 array is running an Enginuity level lower than 5876 Q42012 SR, or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
   b. The SRDF device pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
5. Not allowed if the devices are moving to a group operating in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side

of the new group.

SRDF operations when R2 is target of TimeFinder Clone

Table 58 identifies the allowable SRDF actions when the R2 is the target of a TimeFinder/Clone copy session.

Table 58 Allowable SRDF operations when R2 is the target of a TimeFinder/Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y												Y		
createpair -restore	Y												Y		
createpair -invalidate R1	Y												Y		
createpair -invalidate R2	Y												Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y						Y			Y		Y	Y		
half_swap	Y						Y			Y		Y	Y		
swap -refresh R1	Y						Y			Y		Y	Y		
swap -refresh R2	Y						Y			Y		Y	Y		
establish	Y						Y			Y		Y	Y		
establish -full	Y						Y			Y		Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y						Y			Y		Y	Y		
restore -full	Y						Y			Y		Y	Y		
update	Y						Y			Y		Y	Y		
failback	Y						Y			Y		Y	Y		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y						Y			Y		Y	Y		
failover -restore	Y						Y			Y		Y	Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 58 Allowable SRDF operations when R2 is the target of a TimeFinder/Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
merge	Y						Y			Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y						Y			Y		Y	Y		
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y						Y			Y		Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the TimeFinder/Clone R2 targets.

SRDF operations for Extent-level TimeFinder/Clone sessions

This section lists the allowable SRDF operations for Extent-level TimeFinder /Clone copy sessions on the R1 source and target and the R2 source and target.

SRDF operations when R1 is source of Extent-level Clone

Table 59 identifies the allowable SRDF actions when the R1 is the source of an Extent-level TimeFinder/Clone copy session.

Table 59 Allowable SRDF operations when R1 is the source of an Extent-level Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -invalidate R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -invalidate R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
half_swap	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
swap -refresh R1	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
swap -refresh R2	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
establish -full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
restore -full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
update	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		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		
failover -restore	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y			Y[1]		

Table 59 Allowable SRDF operations when R1 is the source of an Extent-level Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the devices are operating in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the

R1 and either:

a. The SRDF pair is an R22->R1 of a concurrent R2 setup in which either:

   i. The R22 array is running an Enginuity level lower than 5876, or the R1 array is running an Enginuity level lower than 5875.

   ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.

b. The SRDF device pair is not an R22->R1 of a concurrent R2 setup and either:

   i. The Enginuity level is lower than 5875 on either the R1 or R2 array.

   ii. SRDF/A device-level write pacing is not configured for autostart on the R2 side.

SRDF operations when R1 is target of Extent-level Clone

Table 60 identifies the allowable SRDF actions when the R1 is the target of an Extent-level TimeFinder/Clone copy session.

Table 60 Allowable SRDF operations when the R1 is the target of an Extent-level Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y														
createpair -restore	Y														
createpair -invalidate R1	Y														
createpair -invalidate R2	Y														
createpair -format	Y														
deletepair	Y														
half_deletepair	Y														
movepair	Y														
half_movepair	Y														
swap	Y														
half_swap	Y														
swap -refresh R1	Y														
swap -refresh R2	Y														
establish	Y					Y	Y			Y		Y	Y		
establish -full	Y					Y	Y			Y		Y	Y		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y														
restore -full	Y														

Table 60 Allowable SRDF operations when the R1 is the target of an Extent-level Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
update	Y														
failback	Y														
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y														
failover -restore	Y														
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y														
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not _ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y														
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y					Y	Y			Y		Y	Y		
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 60 Allowable SRDF operations when the R1 is the target of an Extent-level Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the Extent-level TimeFinder/Clone R1 targets.

SRDF operations when R2 is source of Extent-level Clone

Table 61 identifies the allowable SRDF actions when the R2 is the source of an Extent-level TimeFinder/Clone copy session.

Table 61 Allowable SRDF operations when R2 is the source of an Extent-level Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[1,3]	Y[3]	Y			Y[3]		
createpair -restore	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y			Y[3]		
createpair -invalidate R1	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y			Y		
createpair -invalidate R2	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y[3]			Y		
createpair -format	Y														
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y[5]	Y[5]	Y[5]	Y	Y	Y	Y[5]	Y[5]	Y	Y	Y	Y[5]		
half_movepair	Y	Y[5]	Y[5]	Y[5]	Y	Y	Y	Y[5]	Y[5]	Y	Y	Y	Y[5]		
swap	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
half_swap	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
swap -refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
establish	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
establish -full	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[1,3]	Y[3]	Y		Y	Y[3]		
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[2]	Y	Y[3]		
restore -full	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[2]	Y	Y[3]		
update	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
failback	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
failover -restore	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 61 Allowable SRDF operations when R2 is the source of an Extent-level Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not _ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y		Y	Y[3]		
rw_disable R2	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[3]	Y	Y[3]		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y[3]	Y[3]	Y[3]	Y	Y	Y	Y[3]	Y[3]	Y	Y[3]	Y	Y[3]		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Only allowed if required Enginuity levels or patches are detected.
2. Not allowed if the devices are in async mode and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair will become an R21->R2 for which any of the following apply:
      i. The R21 array is running an Enginuity level lower than 5876 Q42012 SR, or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
      iii. If the R21->R2 pair will be read/write (RW) on the SRDF link, the R21 must be pace-capable.
   b. The SRDF device pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
3. Not allowed if the devices are in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
4. Not allowed if the devices are moving to a group operating in asynchronous mode and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup and any of the following apply:
      i. The R21 array is running an Enginuity level lower than 5876 Q42012 SR, or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
   b. The SRDF device pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
5. Not allowed if the devices are moving to a group operating in asynchronous mode and R1 and R2 arrays are running Enginuity 5876 or less and there is a TimeFinder/Clone off of the R2 and either:
   a. The SRDF pair is an R21->R2 of a cascaded setup in which either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
   b. The SRDF pair is not an R21->R2 of a cascaded setup and either:
      i. The Enginuity level is lower than 5875 on either the R1 or R2 array.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.

SRDF operations when R2 is target of Extent-level Clone

Table 62 identifies the allowable SRDF actions when the R2 is the target of an Extent-level TimeFinder/Clone copy session.

Table 62 Allowable SRDF operations when the R2 is the target of an Extent-level Clone

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
createpair -establish	Y														
createpair -restore	Y														
createpair -invalidate R1	Y														
createpair -invalidate R2	Y														
createpair -format	Y														
deletepair	Y														
half_deletepair	Y														
movepair	Y														
half_movepair	Y														
swap	Y														
half_swap	Y														
swap -refresh R1	Y														
swap -refresh R2	Y														
establish	Y														
establish -full	Y														
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y														
restore -full	Y														
update	Y														
failback	Y														
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y														
failover -restore	Y														
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 62 Allowable SRDF operations when the R2 is the target of an Extent-level Clone (continued)

SRDF control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
merge	Y														
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
refresh R2	Y														
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y														
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for Extent-level TimeFinder/Clone R2 targets.

SRDF set operations for TimeFinder/Snap sessions

This section lists the allowable SRDF set operations for TimeFinder /Snap copy sessions on the R1 source and target and the R2 source and target.

SRDF set operations when R1 is source of TimeFinder/Snap

Table 63 identifies the allowable SRDF set actions when the R1 is the source of a TimeFinder/Snap copy session.

Table 63 Allowable SRDF set operations when R1 is the source of a TimeFinder/Snap

SRDF set operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R1 is target of TimeFinder/Snap

Table 64 identifies the allowable SRDF set actions when the R1 is the target for a TimeFinder/Snap copy session.

Table 64 Allowable SRDF set operations when R1 is the target of a TimeFinder/Snap

SRDF set operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R2 is source of TimeFinder/Snap

Table 65 identifies the allowable SRDF set actions when the R2 is the source of a TimeFinder/Snap copy session.

Table 65 Allowable SRDF set operations when R2 is the source of a TimeFinder/Snap

SRDF set operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y[1]	Y[1]	Y[1]	Y[1]		Y[1]	Y[1]		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. If the R2 is not an extent-based TimeFinder/Snap source device and R1 and R2 arrays are running Enginuity 5876 or less, then not allowed if either of the following is true:
   a. The RDF device pair is the R21->R2 of a cascaded setup and either:
      i. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 group of the R21 device.
   b. The SRDF pair is not the R21->R2 of a cascaded setup and either:
      i. The R1 or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 group.

SRDF set operations when R2 is target of TimeFinder/Snap

Table 66 identifies the allowable SRDF set when the R2 is the target of a TimeFinder/Snap copy session.

Table 66 Allowable SRDF set operations when R2 is the target of a TimeFinder/Snap

SRDF set operation:	No Session	Create in Progress	Created	Recreated	Copy on Write	Copied	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y										
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations for TimeFinder/Clone sessions

This section lists the allowable SRDF set operations for TimeFinder /Clone copy sessions on the R1 and R2 source and the R1 and R2 target.

SRDF set operations when R1 is source of TimeFinder/Clone

Table 67 identifies the allowable SRDF set actions when the R1 is the source of a TimeFinder/Clone copy session.

Table 67 Allowable SRDF set operations when R1 is the source of a TimeFinder/Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R1 is target of TimeFinder/Clone

Table 68 identifies the allowable SRDF set actions when the R1 is the target of a TimeFinder/Clone copy session.

Table 68 Allowable SRDF set operations when R1 is the target of a TimeFinder/Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y		Y	Y	Y	Y		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if TimeFinder/Clone pair was created with -copy.

SRDF set operations when R2 is source of TimeFinder/Clone

Table 69 identifies the allowable SRDF set actions when the R2 is the source of a TimeFinder/Clone copy session.

Table 69 Allowable SRDF set operations when R2 is the source of a TimeFinder/Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y[1]	Y[1]	Y[1]	Y	Y	Y	Y[1]	Y[1]	Y		Y	Y[1]		
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. If the R2 is not an extent-based TimeFinder/Clone source device and R1 and R2 array are running Enginuity 5876 or less, then not allowed if either of the following is true:
   a. The RDF device pair is the R21->R2 of a cascaded setup and either:
      i. The R21 array is running an Enginuity level lower than 5876 or the R2 array is running an Enginuity level less than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 group of the R21 device.
   b. The SRDF pair is not the R21->R2 of a cascaded setup and either:
      i. The R1 or the R2 array is running an Enginuity level lower than 5875.
      ii. SRDF/A device-level write pacing is not configured for autostart on the R1 group.

SRDF set operations when R2 is target of TimeFinder/Clone

Table 70 identifies the allowable SRDF set actions when the R2 is the target of a TimeFinder/Clone copy session.

Table 70 Allowable SRDF set operations when R2 is the target of a TimeFinder/Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y														
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations for Extent-level TimeFinder/Clone sessions

This section lists the allowable SRDF set operations for Extent-level TimeFinder /Clone copy sessions on the R1 and R2 source and the R1 and R2 target.

SRDF set operations when R1 is source of Extent-level Clone

Table 71 identifies the allowable SRDF set actions when the R1 is the source of an Extent-level TimeFinder/Clone copy session.

Table 71 Allowable SRDF set operations when R1 is the source of an Extent-level Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R1 is target of Extent-level Clone

Table 72 identifies the allowable SRDF set actions when the R1 is the target of an Extent-level TimeFinder/Clone copy session.

Table 72 Allowable SRDF set operations when R1 is the target of an Extent-level Clone

SRDF SET control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R2 is source of Extent-level Clone

Table 73 identifies the allowable SRDF set actions when the R2 is the source of an Extent-level TimeFinder/Clone copy session.

Table 73 Allowable SRDF set operations when the R2 is the source of an Extent-level Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Note: There are no subscripts for the operations in this table.

SRDF set operations when R2 is target of Extent-level Clone

Table 74 identifies the allowable SRDF set actions when the R2 is the target of an Extent-level TimeFinder/Clone copy session.

Table 74 Allowable SRDF set operations when R2 is the target of an Extent-level Clone

SRDF set control operation:	No Session	Create in Progress	Created	Recreated	Precopy	Copy in Progress	Copied	Copy on Write	Copy on Access	Split	Restore in Progress	Restored	Term in Progress	Invalid	Failed
set mode async	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode sync	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

SRDF operations for TimeFinder Snap VX

This section lists the allowable SRDF operations for TimeFinder Snap VX sessions on VMAX 3 arrays running HYPERMAX OS.

Determine SnapVX states-SRDF operations interaction rules

The following table describes how Snap VX states are identified for SRDF operations allowed with Snap VX source devices.

Table 75 SnapVX State Determination

Snap VX State	Snap VX CLI	Determine State
Established	`symsnapvx list`	The flags field indicates: `(F)ailed: = . for No Failure`
Restore In Progress	`symsnapvx list -restored -detail`	The Done(%) field is not at 100%
Restored	`syms‘napvx list -restored -detail`	The Done(%) field is at 100%
Terminate in Progress	`symsnapvx list`	The snapshot is still seen on the display.
Terminated	`symsnapvx list`	The snapshot is not seen on the display.
Failed	`symsnapvx list`	The flags field indicates: `(F)ailed = X for Failed`
Link Copy In Progress	`symsnapvx list -link -tgt -detail`	The flags field indicates: `(C)opy = I for CopyInProg` The Done (%) field is not at 100%
Link Copied	`symsnapvx list -link -tgt -detail`	The flags field indicates: `(C)opy = C for Copied or D for Copied/Destaged` The Done (%) field is at 100%
Linked	`symsnapvx list -linked`	The flags field indicates : `(C)opy = . for NoCopy Link`

SRDF operations when R1 is source of Snapvx

Table 76 identifies allowable SRDF operations when the R1 is the source of a TimeFinder Snapvx session

Table 76 Allowable SRDF operations when R1 is the source of a TimeFinder Snapvx

SRDF Control Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
createpair - establish	Y	Y	Y		Y	Y	
createpair - restore	Y	Y	Y		Y	Y	
createpair - invalidate R1	Y	Y	Y		Y	Y	
createpair - invalidate R2	Y	Y	Y		Y	Y	
createpair - format	Y						
deletepair	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y
swap	Y	Y	Y			Y	
half_swap	Y	Y	Y			Y	
swap -refresh R1	Y	Y	Y			Y	
swap -refresh R2	Y	Y	Y			Y	
establish	Y	Y	Y		Y	Y	
establish -full	Y	Y	Y		Y	Y	
split	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y	Y		Y	Y	
restore -full	Y	Y	Y		Y	Y	
update	Y	Y	Y		Y	Y	
failback	Y	Y	Y		Y	Y	
failover	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y	Y			Y	
failover -restore	Y	Y	Y			Y	
invalidate -R1	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y		Y	Y	

Table 76 Allowable SRDF operations when R1 is the source of a TimeFinder Snapvx

SRDF Control Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
msc_cleanup	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y		Y	Y	
refresh R2	Y	Y	Y	Y	Y	Y	
suspend	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y	Y		Y	Y	
rw_disable R2	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y

SRDF operations when R1 is target for Snapvx

Table 77 identifies allowable SRDF operations when the R1 is the target of a TimeFinder Snapvx session

Table 77 Allowable SRDF operations when R1 is the target of a TimeFinder Snapvx

SRDF Control Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
createpair - establish	Y		Y		
createpair - restore	Y		Y		
createpair - invalidate R1	Y		Y		
createpair - invalidate R2	Y		Y		
createpair - format	Y				
deletepair	Y	Y	Y		Y
half_deletepair	Y	Y	Y		Y
movepair	Y	Y	Y		Y
half_movepair	Y	Y	Y		Y
swap	Y				
half_swap	Y				
swap -refresh R1	Y				
swap -refresh R2	Y				
establish	Y	Y	Y		
establish -full	Y	Y	Y		
split	Y	Y	Y		Y
restore	Y		Y		
restore -full	Y		Y		
update	Y		Y		
failback	Y		Y		
failover	Y	Y	Y		Y
failover -establish	Y				
failover -restore	Y				
invalidate -R1	Y	Y	Y		Y
invalidate -R2	Y	Y	Y		Y
merge	Y		Y		
msc_cleanup	Y	Y	Y		Y

Table 77 Allowable SRDF operations when R1 is the target of a TimeFinder Snapvx

SRDF Control Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
not_ready R1	Y	Y	Y		Y
not_ready R2	Y	Y	Y		Y
ready R1	Y	Y	Y		Y
ready R2	Y	Y	Y		Y
refresh R1	Y		Y		
refresh R2	Y	Y	Y		
suspend	Y	Y	Y		Y
resume	Y		Y		
rw_disable R2	Y	Y	Y		Y
rw_enable R1	Y	Y	Y		Y
rw_enable R2	Y	Y	Y		Y
write_disable R1	Y	Y	Y		Y
write_disable R2	Y	Y	Y		Y
activate -rdfa_dse	Y	Y	Y		Y
deactivate -rdfa_dse	Y	Y	Y		Y
activate -rdfa_devpace	Y	Y	Y		Y
deactivate -rdfa_devpace	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y		Y
deactivate -rdfa_pace	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y		Y
deactivate -rdfa_wpace	Y	Y	Y		Y
activate -rdfa_wpace_exempt	Y	Y	Y		Y
deactivate -rdfa_wpace_exempt	Y	Y	Y		Y

Note: There are no subscripts for the operations in this table.

SRDF operations when R2 is source of Snapvx

Table 78 identifies allowable SRDF operations when the R2 is the source of a TimeFinder Snapvx session

Table 78 Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx

SRDF Control Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
createpair - establish	Y	Y	Y[2]			Y	
createpair - restore	Y	Y	Y			Y	
createpair - invalidate R1	Y	Y	Y			Y	
createpair - invalidate R2	Y	Y	Y			Y	
createpair - format	Y						
deletepair	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	
half_movepair	Y	Y	Y	Y	Y	Y	
swap	Y	Y	Y		Y	Y	
half_swap	Y	Y	Y		Y	Y	
swap -refresh R1	Y	Y	Y		Y	Y	
swap -refresh R2	Y	Y	Y		Y	Y	
establish	Y	Y	Y		Y	Y	
establish -full	Y	Y	Y[2]		Y	Y	
split	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y	Y	Y	Y	Y	
restore -full	Y	Y	Y	Y	Y	Y	
update	Y	Y	Y		Y	Y	
failback	Y	Y	Y		Y	Y	
failover	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y	Y	Y		Y	Y	
failover -restore	Y	Y	Y		Y	Y	
invalidate -R1	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y		Y	Y	
msc_cleanup	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y

Table 78 Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx

SRDF Control Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
ready R1	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	
refresh R2	Y	Y	Y		Y	Y	
suspend	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y	Y		Y	Y	
rw_disable R2	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y

1. If the SRDF/A session is in Transmit Idle state, you must use symforce from the R1 side.
2. Must use -force

SRDF operations when R2 is target for Snapvx

Table 79 identifies allowable SRDF operations when the R2 is the target of a TimeFinder Snapvx session.

Table 79 Allowable SRDF operations when R2 is the target of a TimeFinder Snapvx

SRDF Control Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
createpair - establish	Y				
createpair - restore	Y				
createpair - invalidate R1	Y				
createpair - invalidate R2	Y				
createpair - format	Y				
deletepair	Y	Y	Y		Y
half_deletepair	Y	Y	Y		Y
movepair	Y	Y	Y		Y
half_movepair	Y	Y	Y		Y
swap	Y		Y		
half_swap	Y		Y		
swap -refresh R1	Y		Y		
swap -refresh R2	Y		Y		
establish	Y		Y		
establish -full	Y		Y		
split	Y	Y	Y		Y
restore	Y		Y		
restore -full	Y		Y		
update	Y		Y		
failback	Y		Y		
failover	Y	Y	Y		Y
failover -establish	Y		Y		
failover -restore	Y		Y		
invalidate -R1	Y	Y	Y		Y
invalidate -R2	Y	Y	Y		Y

Table 79 Allowable SRDF operations when R2 is the target of a TimeFinder Snapvx

SRDF Control Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
merge	Y		Y		
msc_cleanup	Y	Y	Y		Y
not_ready R1	Y	Y	Y		Y
not_ready R2	Y	Y	Y		Y
ready R1	Y	Y	Y		Y
ready R2	Y		Y		
refresh R1	Y	Y	Y		
refresh R2	Y	Y	Y		
suspend	Y	Y	Y		Y
resume	Y		Y		
rw_disable R2	Y	Y	Y		Y
rw_enable R1	Y	Y	Y		Y
rw_enable R2	Y	Y	Y		Y
write_disable R1	Y	Y	Y		Y
write_disable R2	Y	Y	Y		Y
activate -rdfa_dse	Y	Y	Y		Y
deactivate -rdfa_dse	Y	Y	Y		Y
activate -rdfa_devpace	Y	Y	Y		Y
deactivate -rdfa_devpace	Y	Y	Y		
activate -rdfa_pace	Y	Y	Y		Y
deactivate -rdfa_pace	Y	Y	Y		
activate -rdfa_wpace	Y	Y	Y		Y
deactivate -rdfa_wpace	Y	Y	Y		Y
activate -rdfa_wpace_exempt	Y	Y	Y		Y
deactivate -rdfa_wpace_exempt	Y	Y	Y		Y

SRDF set operations for TimeFinder Snapvx sessions

This section lists the allowable SRDF set operations for Timefinder Snapvx sessions on the R1 and R2 source, and the R1 and R2 target.

SRDF set operations when R1 is source of TimeFinder Snapvx

Table 80 identifies allowable SRDF set operations when the R1 is the source of a TimeFinder Snapvx session.

Table 80 Allowable SRDF set operations when R1 is the source of a TimeFinder Snapvx

SRDF Set Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
set mode async	Y	Y	Y	Y	Y	Y	
set mode sync	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y

SRDF set operations when R1 is target of TimeFinder Snapvx

Table 81 identifies allowable SRDF set operations when the R1 is the target of a TimeFinder Snapvx session.

Table 81 Allowable SRDF set operations when R1 is the target of a TimeFinder Snapvx

SRDF Set Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
set mode async	Y	Y	Y	Y	
set mode sync	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y

SRDF set operations when R2 is source of TimeFinder Snapvx

Table 82 identifies allowable SRDF set operations when the R2 is the source of a TimeFinder Snapvx session.

Table 82 Allowable SRDF set operations when R2 is the source of a TimeFinder Snapvx

SRDF Set Operation:	No Snapshot	Establish in Progress	Established	Restore in Progress	Restored	Term in Progress	Failed
set mode async	Y	Y	Y		Y	Y	
set mode sync	Y	Y	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y	Y	Y

SRDF set operations when R2 is target of TimeFinder Snapvx

Table 83 identifies allowable SRDF set operations when the R2 is the target of a TimeFinder Snapvx session.

Table 83 Allowable SRDF set operations when R2 is the target of a TimeFinder Snapvx

SRDF Set Operation:	No Link	Link Copy in Progress	Link Copied	Linked	Failed
set mode async	Y				
set mode sync	Y	Y	Y	Y	Y
set mode acp_disk	Y	Y	Y	Y	Y
set mode acp_wp	Y	Y	Y	Y	Y

APPENDIX C
SRDF operations and rcopy States

This appendix describes the following topics:

- R1 is part of an Rcopy PUSH 448
- R1 is part of an Rcopy PULL 450
- R2 is part of an Rcopy PUSH 453
- R2 is part of an Rcopy PULL 455

Rcopy Session on the R1 side

This section lists the allowable SRDF operations with their applicable Rcopy states when there is an Rcopy session on the R1.

R1 is part of an Rcopy PUSH

Table 84 identifies the allowable SRDF actions with their applicable Rcopy states when there is an Rcopy PUSH session on the R1.

Table 84 Allowable SRDF operations with Rcopy PUSH session on the R1

Rcopy State: / SRDF Control Operation:	None	Create in Progress	Created	Copy in Progress	Copy on Write	Copied	Recreate in Progress	Recreated	Terminate in Progress	Failed	Invalid	Verify In Progress	Restored	Restored In Prog	Precopy	Sync In Prog	Synchronised	Stopped
createpair -establish	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y [7]	Y[1,6,7]	Y [7]	Y [7]	Y [7]	Y [7]
createpair -restore	Y		Y[1,7]			Y[1,7]		Y[1,7]					Y[1,6,7]					
createpair -invalidate R1	Y		Y[1,7]			Y[1,7]		Y[1,7]					Y[1,6,7]					
createpair -invalidate R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[1,6,7]	Y	Y	Y	Y	Y
createpair -format	Y																	
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y																	
half_swap	Y																	
swap -refresh R1	Y																	
swap -refresh R2	Y																	
establish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[4]	Y[1,6]	Y	Y	Y	Y
establish -full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[4]	Y[1,6]	Y	Y	Y	Y
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					
restore -full	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					
update R1	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					
failback	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					

Table 84 Allowable SRDF operations with Rcopy PUSH session on the R1 (continued)

Rcopy State: / SRDF Control Operation:	None	Create in Progress	Created	Copy in Progress	Copy on Write	Copied	Recreate in Progress	Recreated	Terminate in Progress	Failed	Invalid	Verify In Progress	Restored	Restored In Prog	Precopy	Sync In Prog	Synchronised	Stopped
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y																	
failover -restore	Y																	
invalidate -R1	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[6]	Y[1,6]	Y	Y	Y	Y
merge	Y	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y		Y[1]			Y[1]		Y[1]					Y[1,6]					
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[6]	Y[1,6]	Y	Y	Y	Y
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y[2]	Y[5]	Y[2]	Y[2]	Y[5]	Y[2]	Y[5]	Y[2]	Y[2]	Y[2]	Y[2]	Y[5,6]	Y[2,6]	Y[2]	Y[2]	Y[2]	Y[2]
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 84 Allowable SRDF operations with Rcopy PUSH session on the R1 (continued)

Rcopy State: / SRDF Control Operation:	None	Create in Progress	Created	Copy in Progress	Copy on Write	Copied	Recreate in Progress	Recreated	Terminate in Progress	Failed	Invalid	Verify In Progress	Restored	Restored In Prog	Precopy	Sync In Prog	Synchronised	Stopped
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the R1 array is running an Enginuity level lower than 5874.
2. Not allowed if the R2 owes data to the R1.
3. If the R2 owes data to the R1, not allowed if the R1 array is running an Enginuity level lower than 5874, or if donor update is specified.
4. Not allowed if the Rcopy session has front end zero detect.
5. If R2 owes data to the R1, not allowed if the R1 array is running an Enginuity level less than 5874 or if donor update specified.
6. Not allowed if the Rcopy session has front end zero detect.
7. Not allowed if the R1 array is running HYPERMAX OS (5977) or above.

R1 is part of an Rcopy PULL

Table 85 identifies the allowable SRDF actions with their applicable Rcopy states when there is an Rcopy PULL session on the R1.

Table 85 Allowable SRDF operations with Rcopy PULL session on the R1

Rcopy State: / SRDF Control Operations:	None	Create in Progress	Created	Copy In Progress	Copy on Access	Copied	Terminate In Progress	Failed	Invalid	Verify In Progress	Sync In Prog	Synchronised	Stopped	Failback
createpair -establish	Y													
createpair -restore	Y													
createpair -invalidate R1	Y													
createpair -invalidate R2	Y													
createpair -format	Y													
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 85 Allowable SRDF operations with Rcopy PULL session on the R1 (continued)

Rcopy State: / SRDF Control Operations:	None	Create in Progress	Created	Copy In Progress	Copy on Access	Copied	Terminate In Progress	Failed	Invalid	Verify In Progress	Sync In Prog	Synchronised	Stopped	Failback
swap	Y													
half_swap	Y													
swap -refresh R1	Y													
swap -refresh R2	Y													
establish	Y	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]
establish-full	Y	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]	Y[1,3]
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y													
restore -full	Y													
update	Y													
failback	Y													
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y													
failover -restore	Y													
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]	Y[2]
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y													
refresh R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y	Y[1,2,3]	Y[1,3]	Y[1,2,3]	Y[1,2,3]	Y[1,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]	Y[1,2,3]
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 85 Allowable SRDF operations with Rcopy PULL session on the R1 (continued)

Rcopy State: / SRDF Control Operations:	None	Create in Progress	Created	Copy In Progress	Copy on Access	Copied	Terminate In Progress	Failed	Invalid	Verify In Progress	Sync In Prog	Synchronised	Stopped	Failback
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the R1 array is running an Enginuity level lower than 5874.
2. Not allowed if the R2 owes data to the R1.
3. Not allowed if the Rcopy session has front end zero detect.
4. If R2 owes data to the R1, not allowed if the R1 array is running an Enginuity level less than 5874.

Rcopy Session on the R2 side

This section lists the allowable SRDF operations with their applicable Rcopy states when there is an Rcopy session on the R2.

R2 is part of an Rcopy PUSH

Table 86 identifies the allowable SRDF actions with their applicable Rcopy states when there is an Rcopy PUSH session on the R2.

Table 86 Allowable SRDF operations with Rcopy PUSH session on the R2

Rcopy state: / SRDF control operation:	None	Create in Progress	Created	Copy In Progress	Copy on Write	Copied	Recreate In Progress	Recreated	Terminate In Progress	Failed	Invalid	Verify In Progress	Restored	Restored In Prog	Precopy	Sync In Prog	Synchronised	Stopped
createpair -establish	Y																	
createpair -restore	Y																	
createpair -invalidate R1	Y																	
createpair -invalidate R2	Y																	
createpair -format	Y																	
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y																	
half_swap	Y																	
swap -refresh R1	Y																	
swap -refresh R2	Y																	
establish	Y		Y[1]			Y[1]		Y[1]					Y[1,2,3]					
establish -full	Y		Y[1]			Y[1]		Y[1]					Y[1,2,3]					
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,3]	Y[3]	Y[1]	Y[1]	Y[1]	Y[1]
restore -full	Y	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1]	Y[1,3]	Y[3]	Y[1]	Y[1]	Y[1]	Y[1]
update	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y[3]	N	Y	Y	Y	Y
failback	Y																	
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y																	
failover -restore	Y																	
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 86 Allowable SRDF operations with Rcopy PUSH session on the R2 (continued)

Rcopy state: / SRDF control operation:	None	Create in Progress	Created	Copy In Progress	Copy on Write	Copied	Recreate In Progress	Recreated	Terminate In Progress	Failed	Invalid	Verify In Progress	Restored	Restored In Prog	Precopy	Sync In Prog	Synchronised	Stopped
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R2	Y																	
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y		Y[1]			Y[1]		Y[1]					Y[1,2,3]					
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

1. Not allowed if the R2 array is running an Enginuity level lower than 5874.
2. Not allowed if donor update specified.

3. Not allowed if the Rcopy session has front end zero detect.

R2 is part of an Rcopy PULL

Table 87 identifies the allowable SRDF actions with their applicable Rcopy states when there is an Rcopy PULL session on the R2.

Table 87 Allowable SRDF operations with Rcopy PULL session on the R2

Rcopy State: / SRDF control operation:	None	Create in Progress	Created	Copy in Progress	Copy on Access	Copied	Terminate In Progress	Failed	Invalid	Verify In Progress	Sync In Prog	Synchronised	Stopped	Failback
createpair -establish	Y													
createpair -restore	Y													
createpair -invalidate R1	Y													
createpair -invalidate R2	Y													
createpair -format	Y													
deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_deletepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
half_movepair	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
swap	Y													
half_swap	Y													
swap -refresh R1	Y													
swap -refresh R2	Y													
establish	Y													
establish -full	Y													
split	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
restore	Y													
restore -full	Y													
update	Y													
failback	Y													
failover	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
failover -establish	Y													
failover -restore	Y													
invalidate -R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 87 Allowable SRDF operations with Rcopy PULL session on the R2

Rcopy State: / SRDF control operation:	None	Create in Progress	Created	Copy in Progress	Copy on Access	Copied	Terminate In Progress	Failed	Invalid	Verify In Progress	Sync In Prog	Synchronised	Stopped	Failback
invalidate -R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
merge	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
msc_cleanup	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
not_ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ready R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refresh R2	Y													
suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
resume	Y													
rw_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
rw_enable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
write_disable R2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_dse	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_devpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_pace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
activate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
deactivate -rdfa_wpace_exempt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y