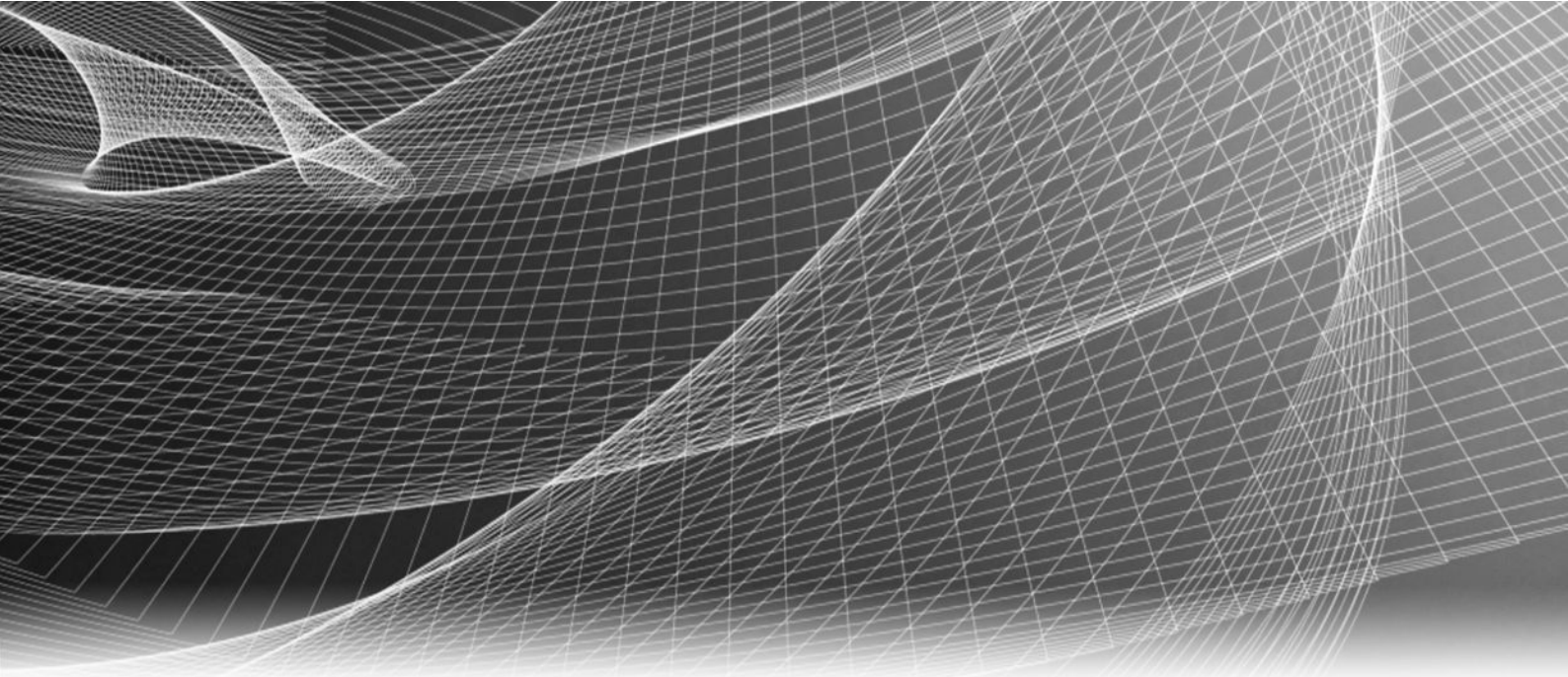




# Solutions Enabler SRDF Family

Version 8.1

## CLI User Guide

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

Published September, 2015

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

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

EMC<sup>2</sup>, EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

EMC Corporation  
Hopkinton, Massachusetts 01748-9103  
1-508-435-1000 In North America 1-866-464-7381  
[www.EMC.com](http://www.EMC.com)

# CONTENTS

|                         |                                                                  |
|-------------------------|------------------------------------------------------------------|
| <b>Figures</b>          | <b>13</b>                                                        |
| <b>Tables</b>           | <b>17</b>                                                        |
| <b>PREFACE</b>          | <b>19</b>                                                        |
| <b>Revision history</b> | <b>21</b>                                                        |
| <b>Chapter 1</b>        | <b>SRDF CLI overview 25</b>                                      |
|                         | Introduction to SRDF.....26                                      |
|                         | HYPERMAX OS.....27                                               |
|                         | Enginuity versions 5773 - 5876.....27                            |
|                         | SRDF documentation .....27                                       |
|                         | What's new in Solutions Enabler 8.1 .....27                      |
|                         | SRDF restrictions .....28                                        |
|                         | SRDF backward compatibility to Enginuity 5876 .....28            |
|                         | SYMCLI for SRDF .....30                                          |
|                         | SYMCLI command syntax .....30                                    |
|                         | Get command help.....30                                          |
|                         | Set environmental variables.....31                               |
|                         | Preset names and IDs.....31                                      |
|                         | SYMCLI SRDF commands .....31                                     |
|                         | symrdf command options.....33                                    |
|                         | Commands to display, query and verify SRDF configurations.....36 |
|                         | symrdf list command options.....40                               |
|                         | ping command.....42                                              |
|                         | verify command .....42                                           |
|                         | SRDF pair states and links.....44                                |
|                         | SRDF pair states.....45                                          |
|                         | Before you begin .....48                                         |
|                         | Array access rights .....49                                      |
|                         | Device external locks .....49                                    |
|                         | SRDF operations and copy sessions .....49                        |
|                         | Mirror R1 to a larger R2 device .....49                          |
|                         | Restrict synchronization .....50                                 |
|                         | SRDF software and hardware compression .....50                   |
|                         | Set compression for SRDF.....50                                  |
|                         | SRDF/A and the consistency exempt option .....51                 |
|                         | Mixed-mode workloads on an SRDF director .....51                 |
|                         | Set mixed-mode workloads.....52                                  |
|                         | FAST VP SRDF coordination .....52                                |
| <b>Chapter 2</b>        | <b>Basic SRDF Control Operations 53</b>                          |
|                         | Basic SRDF control operations summary .....54                    |
|                         | SRDF basic control operations.....56                             |
|                         | SRDF modes of operation .....56                                  |

|                  |                                                   |            |
|------------------|---------------------------------------------------|------------|
|                  | Establish an SRDF pair (full) .....               | 60         |
|                  | Establish an SRDF pair (incremental) .....        | 62         |
|                  | Failback to source .....                          | 64         |
|                  | Failover to target .....                          | 66         |
|                  | Invalidate R1 tracks .....                        | 68         |
|                  | Invalidate R2 tracks .....                        | 68         |
|                  | Make R1 ready .....                               | 69         |
|                  | Make R1 not ready .....                           | 70         |
|                  | Make R2 ready .....                               | 70         |
|                  | Make R2 not ready .....                           | 71         |
|                  | Merge track tables .....                          | 71         |
|                  | Move one-half of an SRDF pair .....               | 72         |
|                  | Move both sides of SRDF device pairs .....        | 72         |
|                  | Read/write disable target device .....            | 73         |
|                  | Refresh R1 .....                                  | 74         |
|                  | Refresh R2 .....                                  | 74         |
|                  | Restore SRDF pairs (full) .....                   | 75         |
|                  | Restore SRDF pairs (incremental) .....            | 77         |
|                  | Resume I/O on links .....                         | 79         |
|                  | Split .....                                       | 80         |
|                  | Suspend I/O on links .....                        | 83         |
|                  | Swap one-half of an SRDF pair .....               | 84         |
|                  | Swap SRDF pairs .....                             | 84         |
|                  | Update R1 mirror .....                            | 84         |
|                  | Write disable R1 .....                            | 86         |
|                  | Write disable R2 .....                            | 87         |
|                  | Write enable R1 .....                             | 87         |
|                  | Write enable R2 .....                             | 88         |
| <b>Chapter 3</b> | <b>Dynamic Operations</b>                         | <b>89</b>  |
|                  | Dynamic operations overview .....                 | 90         |
|                  | SRDF group maximum per OS .....                   | 90         |
|                  | HYPERMAX OS and SRDF groups .....                 | 90         |
|                  | SRDF group attributes .....                       | 91         |
|                  | Manage SRDF groups .....                          | 92         |
|                  | Create an SRDF group and add pairs .....          | 92         |
|                  | Modifying dynamic SRDF groups .....               | 96         |
|                  | Removing dynamic SRDF groups .....                | 99         |
|                  | Device pairing operations .....                   | 101        |
|                  | Create a device file .....                        | 101        |
|                  | Supported device types for SRDF pairs .....       | 101        |
|                  | Block createpair when R2 is larger than R1 .....  | 102        |
|                  | Creating SRDF device pairs .....                  | 102        |
|                  | Create dynamic concurrent pairs .....             | 112        |
|                  | Deleting dynamic SRDF device pairs .....          | 113        |
|                  | Group, move and swap dynamic devices .....        | 116        |
|                  | Creating a device group using a device file ..... | 116        |
|                  | Move dynamic SRDF device pairs .....              | 117        |
|                  | Swapping SRDF devices .....                       | 118        |
|                  | Dynamic failover operations .....                 | 121        |
| <b>Chapter 4</b> | <b>SRDF/Asynchronous Operations</b>               | <b>125</b> |
|                  | SRDF/Asynchronous operations overview .....       | 126        |
|                  | SRDF/A restrictions .....                         | 126        |

|                  |                                                                   |            |
|------------------|-------------------------------------------------------------------|------------|
|                  | SRDF/A cycle modes .....                                          | 127        |
|                  | Protect the R2 side with TimeFinder BCVs .....                    | 129        |
|                  | Drop SRDF/A session immediately.....                              | 129        |
|                  | SRDF/Asynchronous operations.....                                 | 129        |
|                  | Transition replication modes .....                                | 130        |
|                  | Set SRDF/A group cycle time, priority, and transmit idle .....    | 132        |
|                  | Check for R1 invalid tracks .....                                 | 133        |
|                  | Consistency for SRDF/A devices .....                              | 134        |
|                  | Add/remove devices with the consistency exempt option .....       | 135        |
|                  | Adding device pairs to an active SRDF/A session .....             | 136        |
|                  | Removing device pairs from an active SRDF/A session .....         | 137        |
|                  | Display checkpoint complete status .....                          | 138        |
|                  | Delta Set Extension management.....                               | 139        |
|                  | DSE SRP capacity management (HYPERMAX OS) .....                   | 139        |
|                  | DSE pool management (Engenuity 5773 - 5876) .....                 | 141        |
|                  | Manage transmit idle .....                                        | 147        |
|                  | Manage SRDF/A write pacing .....                                  | 147        |
|                  | Devices that cannot be paced in a cascaded SRDF configuration ... | 150        |
|                  | Set SRDF/A group-level write pacing attributes .....              | 151        |
|                  | Activate write pacing .....                                       | 153        |
|                  | Simultaneous group-level and device-level write pacing .....      | 153        |
|                  | Display SRDF/A .....                                              | 154        |
|                  | Show SRDF/A group information .....                               | 154        |
|                  | List SRDF/A- capable devices .....                                | 154        |
| <b>Chapter 5</b> | <b>SRDF/Metro Operations</b>                                      | <b>157</b> |
|                  | SRDF/Metro Overview.....                                          | 158        |
|                  | Witness array option.....                                         | 160        |
|                  | Witness SRDF groups.....                                          | 161        |
|                  | Add an SRDF/Metro Witness group .....                             | 161        |
|                  | Remove an SRDF/Metro Witness group.....                           | 161        |
|                  | Modify an SRDF/Metro Witness group.....                           | 162        |
|                  | Suspend an SRDF/Metro group.....                                  | 162        |
|                  | Create device pairs in SRDF/Metro configurations.....             | 162        |
|                  | Create SRDF/Metro pairs with the -establish option.....           | 163        |
|                  | Create SRDF/Metro pairs with the -invalidate option.....          | 164        |
|                  | Create SRDF/Metro pairs with the -restore option.....             | 164        |
|                  | SRDF/Metro CLI operations supported and restrictions.....         | 166        |
|                  | Setting up SRDF/Metro (Witness array option).....                 | 168        |
|                  | Setting SRDF/Metro bias .....                                     | 173        |
|                  | Deactivate SRDF/Metro.....                                        | 175        |
|                  | Restoring native device personality .....                         | 175        |
|                  | Display SRDF/Metro.....                                           | 176        |
| <b>Chapter 6</b> | <b>Consistency Group Operations</b>                               | <b>181</b> |
|                  | Consistency group operations overview.....                        | 182        |
|                  | Consistency protection using the SRDF daemon .....                | 182        |
|                  | Redundant consistency protection .....                            | 183        |
|                  | SRDF consistency group operations.....                            | 184        |
|                  | Creating a consistency group .....                                | 185        |
|                  | Create composite groups from various sources.....                 | 186        |
|                  | Enable and disable SRDF consistency protection.....               | 188        |
|                  | Enable consistency: composite group vs. SRDF group name .....     | 189        |

|                                 |                                                                                                                          |            |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------|------------|
|                                 | Enabling SRDF consistency protection for concurrent SRDF devices .....                                                   | 191        |
|                                 | Check if device pairs are enabled for consistency protection .....                                                       | 193        |
|                                 | Block symcg enable on R2 side .....                                                                                      | 194        |
|                                 | Delete an SRDF consistency group .....                                                                                   | 194        |
|                                 | Suspend SRDF consistency protection .....                                                                                | 195        |
|                                 | Composite group cleanup (msc_cleanup) .....                                                                              | 196        |
| Modify consistency groups ..... |                                                                                                                          | 197        |
|                                 | Before you begin consistency group modification .....                                                                    | 197        |
|                                 | Consistency group modification restrictions .....                                                                        | 198        |
|                                 | Prepare staging area for consistency group modification .....                                                            | 198        |
|                                 | Restrictions: Add devices to SRDF consistency group .....                                                                | 201        |
|                                 | Restrictions: Remove devices from SRDF consistency group .....                                                           | 202        |
|                                 | Restrictions: Device types allowed for add operations to an RDF1 consistency group .....                                 | 202        |
|                                 | Restrictions: Device types and consistency modes allowed for add operations to a concurrent RDF1 consistency group ..... | 203        |
|                                 | Restrictions: Devices types allowed to add to a cascaded RDF1 consistency group .....                                    | 204        |
|                                 | Restrictions: Device types allowed for remove operations from an RDF1 consistency group .....                            | 206        |
|                                 | Restrictions: Device types allowed for remove operations from a concurrent RDF1 consistency group .....                  | 206        |
|                                 | Restrictions: Device types allowed for remove operations from a cascaded RDF1 consistency group .....                    | 207        |
|                                 | Recovering from a failed dynamic modify operation .....                                                                  | 207        |
|                                 | Consistency groups with a parallel database .....                                                                        | 207        |
|                                 | Consistency groups with BCV access at the target site .....                                                              | 208        |
| <b>Chapter 7</b>                | <b>Concurrent Operations</b>                                                                                             | <b>211</b> |
|                                 | Concurrent operations overview .....                                                                                     | 212        |
|                                 | Concurrent operations restrictions .....                                                                                 | 212        |
|                                 | Additional documentation for concurrent operations .....                                                                 | 214        |
|                                 | Configuring a concurrent SRDF relationship .....                                                                         | 214        |
|                                 | Creating and establishing concurrent SRDF devices .....                                                                  | 214        |
|                                 | Split concurrent SRDF devices .....                                                                                      | 215        |
|                                 | Restore concurrent devices .....                                                                                         | 216        |
|                                 | View concurrent SRDF devices .....                                                                                       | 218        |
| <b>Chapter 8</b>                | <b>Cascaded Operations</b>                                                                                               | <b>221</b> |
|                                 | Cascaded operations overview .....                                                                                       | 222        |
|                                 | SRDF modes in cascaded configurations .....                                                                              | 223        |
|                                 | SRDF modes in cascaded configurations with EDP .....                                                                     | 224        |
|                                 | Restrictions: Cascaded operations .....                                                                                  | 224        |
|                                 | Setting up cascaded SRDF .....                                                                                           | 225        |
|                                 | Setting up a relationship for cascaded SRDF .....                                                                        | 225        |
|                                 | Applicable pair states for cascaded SRDF operations .....                                                                | 227        |
|                                 | RDF21 SRDF groups .....                                                                                                  | 227        |
|                                 | R21 device management .....                                                                                              | 228        |
|                                 | Hop 2 controls in cascaded SRDF .....                                                                                    | 228        |
|                                 | Cascaded SRDF with EDP .....                                                                                             | 230        |
|                                 | SRDF/EDP restrictions .....                                                                                              | 230        |
|                                 | Setting up cascaded SRDF with EDP .....                                                                                  | 231        |
|                                 | Restrictions for diskless devices in cascaded SRDF .....                                                                 | 233        |

|                  |                                                                                    |            |
|------------------|------------------------------------------------------------------------------------|------------|
|                  | Create diskless devices .....                                                      | 234        |
|                  | Add a diskless SRDF mirror .....                                                   | 235        |
|                  | Restart a diskless configuration .....                                             | 236        |
|                  | Sample session: planned failover .....                                             | 236        |
|                  | Display cascaded SRDF.....                                                         | 238        |
|                  | List cascaded SRDF devices .....                                                   | 238        |
|                  | Diskless devices.....                                                              | 239        |
|                  | Query hop 2 information .....                                                      | 241        |
| <b>Chapter 9</b> | <b>SRDF/Star Operations</b>                                                        | <b>245</b> |
|                  | SRDF/Star operations overview.....                                                 | 246        |
|                  | Cascaded SRDF/Star .....                                                           | 246        |
|                  | Concurrent SRDF/Star .....                                                         | 247        |
|                  | Concurrent SRDF/Star with R22 devices.....                                         | 248        |
|                  | SRDF/Star features.....                                                            | 250        |
|                  | SRDF/Star restrictions .....                                                       | 250        |
|                  | SRDF/Star states and operations.....                                               | 251        |
|                  | SRDF/Star state.....                                                               | 251        |
|                  | Target site states .....                                                           | 251        |
|                  | SRDF/Star site configuration transitions .....                                     | 252        |
|                  | SRDF/Star operation categories.....                                                | 253        |
|                  | Required states for operations: Concurrent SRDF/Star.....                          | 254        |
|                  | Required states for operations: Cascaded SRDF/Star.....                            | 258        |
|                  | SRDF/Star operations summary .....                                                 | 263        |
|                  | symstar command options .....                                                      | 264        |
|                  | Command failure while in Connected state .....                                     | 267        |
|                  | Restrictions for cascaded mode.....                                                | 267        |
|                  | Configure and bring up SRDF/Star .....                                             | 267        |
|                  | Step 1: Verify SRDF/Star host connectivity .....                                   | 268        |
|                  | Step 2: Verify array settings .....                                                | 269        |
|                  | Step 3: Create an SRDF/Star composite group .....                                  | 269        |
|                  | Step 4: Create the SRDF/Star options file .....                                    | 274        |
|                  | Step 5: Perform the symstar setup operation .....                                  | 276        |
|                  | Step 6: Create R2 or R21 composite groups .....                                    | 277        |
|                  | Step 7: (Optional) Add BCV devices to the SRDF/Star configuration .....            | 278        |
|                  | Step 8: Bring up the SRDF/Star configuration.....                                  | 279        |
|                  | Displaying the symstar configuration .....                                         | 280        |
|                  | Basic SRDF/Star operations .....                                                   | 283        |
|                  | Isolate SRDF/Star sites .....                                                      | 284        |
|                  | Unprotect target sites.....                                                        | 285        |
|                  | Halt target sites.....                                                             | 285        |
|                  | Clean up metadata .....                                                            | 285        |
|                  | SRDF/Star consistency group operations .....                                       | 286        |
|                  | Before you begin: SRDF daemon interaction .....                                    | 286        |
|                  | SRDF/Star consistency group restrictions.....                                      | 287        |
|                  | Prepare staging for SRDF/Star consistency group modification.....                  | 287        |
|                  | Add devices to a concurrent SRDF/Star consistency group .....                      | 288        |
|                  | Add devices to a cascaded SRDF/Star consistency group .....                        | 292        |
|                  | Remove devices from consistency groups.....                                        | 294        |
|                  | Recovering from a failed consistency group modification .....                      | 296        |
|                  | Recovery operations: Concurrent SRDF/Star .....                                    | 297        |
|                  | Recover from transient faults: concurrent SRDF/Star.....                           | 297        |
|                  | Recover from a transient fault without reconfiguration: concurrent SRDF/Star ..... | 298        |

|                   |                                                                                    |            |
|-------------------|------------------------------------------------------------------------------------|------------|
|                   | Recover from transient fault with reconfiguration: concurrent SRDF/Star .....      | 299        |
|                   | Recover using reconfigure operations .....                                         | 300        |
|                   | Workload switching: Concurrent SRDF/Star .....                                     | 301        |
|                   | Planned workload switching: Concurrent SRDF/Star .....                             | 302        |
|                   | Unplanned workload switching: concurrent SRDF/Star.....                            | 306        |
|                   | Unplanned workload switch to synchronous target site: concurrent SRDF/Star .....   | 307        |
|                   | Unplanned workload switch to asynchronous target site: concurrent SRDF/Star .....  | 312        |
|                   | Switch back to the original workload site: concurrent SRDF/Star ...                | 316        |
|                   | Recovery operations: Cascaded SRDF/Star .....                                      | 318        |
|                   | Recovering from transient faults: Cascaded SRDF/Star .....                         | 318        |
|                   | Recovering from transient faults without reconfiguration: Cascaded SRDF/Star ..... | 319        |
|                   | Recovering from transient faults with reconfiguration: Cascaded SRDF/Star .....    | 320        |
|                   | Workload switching: Cascaded SRDF/Star .....                                       | 322        |
|                   | Planned workload switching: Cascaded SRDF/Star .....                               | 322        |
|                   | Unplanned workload switching: cascaded SRDF/Star .....                             | 324        |
|                   | Reconfiguration operations .....                                                   | 333        |
|                   | Before you begin reconfiguration operations.....                                   | 334        |
|                   | Reconfiguring mode: cascaded to concurrent .....                                   | 334        |
|                   | Reconfiguring cascaded paths.....                                                  | 338        |
|                   | Reconfiguring mode: concurrent to cascaded .....                                   | 340        |
|                   | Reconfigure mode without halting the workload site .....                           | 344        |
|                   | SRDF/Star configuration with R22 devices .....                                     | 345        |
|                   | Before you begin SRDF/Star configuration with R22 devices.....                     | 345        |
|                   | Transition SRDF/Star to use R22 devices .....                                      | 346        |
| <b>Chapter 10</b> | <b>Device Migration Operations</b>                                                 | <b>349</b> |
|                   | Device Migration operations overview .....                                         | 350        |
|                   | Device Migration operations requirements .....                                     | 350        |
|                   | R1 device migration .....                                                          | 351        |
|                   | Configure a temporary SRDF group .....                                             | 351        |
|                   | Establish a concurrent SRDF relationship .....                                     | 352        |
|                   | Replacing the R1 device .....                                                      | 352        |
|                   | R2 device migration .....                                                          | 353        |
|                   | Configure setup for R2 migration .....                                             | 354        |
|                   | Establish a concurrent SRDF relationship .....                                     | 355        |
|                   | Replacing the R2 device .....                                                      | 356        |
|                   | R1 and R2 migration procedures.....                                                | 357        |
|                   | Before you begin R1 and R2 migration.....                                          | 357        |
|                   | Restrictions for R1 and R2 migration.....                                          | 358        |
|                   | Sample procedure: migrating R1 devices .....                                       | 359        |
|                   | Sample procedure: migrating R2 devices .....                                       | 365        |
|                   | SRDF pair states for migration .....                                               | 368        |
|                   | Pair states for migrate -setup .....                                               | 368        |
|                   | Pair states for migrate -replace for first leg of concurrent SRDF .....            | 370        |
|                   | Pair states for migrate -replace for second leg of concurrent SRDF ..              | 372        |
| <b>Chapter 11</b> | <b>SRDF/Automated Replication</b>                                                  | <b>375</b> |
|                   | SRDF/Automated Replication overview.....                                           | 376        |
|                   | Restrictions: SRDF/Automated Replication.....                                      | 376        |



|                   |                                                                                     |            |
|-------------------|-------------------------------------------------------------------------------------|------------|
|                   | SRDF/Automated Replication operations .....                                         | 377        |
|                   | Configure single-hop sessions .....                                                 | 377        |
|                   | Setting up single-hop data replication .....                                        | 377        |
|                   | Setting up single hop manually .....                                                | 380        |
|                   | Configure multi-hop sessions .....                                                  | 381        |
|                   | Concurrent BCVs with SRDF/AR .....                                                  | 384        |
|                   | Setting replication cycle parameters .....                                          | 384        |
|                   | Clustered SRDF/AR.....                                                              | 386        |
|                   | Write log files to a specified SFS .....                                            | 387        |
|                   | Restart from another host .....                                                     | 387        |
|                   | List log files written to the SFS .....                                             | 388        |
|                   | Show log files written to SFS .....                                                 | 388        |
|                   | Delete a log file written to SFS .....                                              | 389        |
|                   | Set symreplicate parameters in the options file.....                                | 389        |
|                   | Format of the symreplicate options file .....                                       | 389        |
|                   | Set replication retry and sleep times .....                                         | 390        |
|                   | Setting the symreplicate control parameters .....                                   | 390        |
|                   | Manage locked devices .....                                                         | 394        |
|                   | Recover locks .....                                                                 | 394        |
|                   | Release locks.....                                                                  | 394        |
|                   | Acquire persistent locks .....                                                      | 395        |
| <b>Chapter 12</b> | <b>TimeFinder and SRDF operations</b>                                               | <b>397</b> |
|                   | TimeFinder consistent splits in SRDF configurations.....                            | 398        |
|                   | Enginuity Consistency Assist .....                                                  | 398        |
|                   | Multi-hop operations .....                                                          | 399        |
|                   | Before you begin: preparing for multi-hop operations .....                          | 399        |
|                   | Control basic operations in a multi-hop configuration .....                         | 400        |
|                   | System-wide split commands.....                                                     | 402        |
|                   | TimeFinder SnapVX and SRDF.....                                                     | 403        |
|                   | TimeFinder SnapVX and Cascaded SRDF.....                                            | 404        |
|                   | TimeFinder SnapVX and Concurrent SRDF.....                                          | 405        |
| <b>Chapter 13</b> | <b>SRDF Automated Recovery Operations</b>                                           | <b>407</b> |
|                   | Automated Recovery overview.....                                                    | 408        |
|                   | SRDF Automated Recovery restrictions.....                                           | 409        |
|                   | Launch SRDF Automated Recovery.....                                                 | 410        |
|                   | Recover cascaded SRDF.....                                                          | 412        |
|                   | Stop SRDF Automated Recovery.....                                                   | 413        |
|                   | symrecover options file parameters .....                                            | 413        |
| <b>Appendix A</b> | <b>SRDF operations and pair states</b>                                              | <b>419</b> |
|                   | SRDF operations and applicable pair states .....                                    | 420        |
|                   | Control operations for R1 - R2 pair states .....                                    | 420        |
|                   | Cascaded SRDF control operations and applicable pair states.....                    | 424        |
|                   | Cascaded SRDF: R1 - R21 control operations allowed for R21- R2 pair<br>states.....  | 424        |
|                   | Cascaded SRDF: R21 - R2 control operations allowed for R1 - R21 pair<br>states..... | 426        |
|                   | Cascaded SRDF set operations and applicable pair states.....                        | 430        |
|                   | Cascaded SRDF: R1 - R21 set operations allowed for R21 - R2 pair<br>states.....     | 430        |

|                   |                                                                               |            |
|-------------------|-------------------------------------------------------------------------------|------------|
|                   | Cascaded SRDF: R21 - R2 set operations allowed for R1 - R21 pair states ..... | 430        |
|                   | Concurrent SRDF operations and applicable pair states .....                   | 432        |
|                   | About concurrent R1 .....                                                     | 432        |
|                   | Concurrent SRDF: 1st leg R1- 2nd leg R1 pair states .....                     | 432        |
|                   | Concurrent SRDF: 1st leg R2 - 2nd leg R2 pair states .....                    | 435        |
|                   | Consistency group operations and applicable pair states .....                 | 438        |
| <b>Appendix B</b> | <b>SRDF operations and TimeFinder sessions</b>                                | <b>439</b> |
|                   | SRDF operations during TimeFinder/Snap/VP Snap sessions .....                 | 440        |
|                   | SRDF operations when R1 is source of TimeFinder Snap/VP Snap ..               | 440        |
|                   | SRDF operations when R1 is target of TimeFinder Snap/VP Snap ...              | 442        |
|                   | SRDF operations when R2 is source of TimeFinder Snap/VP Snap ...              | 445        |
|                   | SRDF operations when R2 is target of TimeFinder Snap/VP Snap ...              | 447        |
|                   | SRDF set operations for TimeFinder/Snap sessions .....                        | 450        |
|                   | SRDF set operations when R1 is source of TimeFinder/Snap .....                | 450        |
|                   | SRDF set operations when R1 is target of TimeFinder/Snap .....                | 450        |
|                   | SRDF set operations when R2 is source of TimeFinder/Snap .....                | 451        |
|                   | SRDF set operations when R2 is target of TimeFinder/Snap .....                | 451        |
|                   | SRDF operations for TimeFinder/Clone sessions .....                           | 453        |
|                   | SRDF operations when R1 is source of TimeFinder Clone .....                   | 453        |
|                   | SRDF operations when R1 is target of TimeFinder Clone .....                   | 455        |
|                   | SRDF operations when R2 is source of TimeFinder Clone .....                   | 457        |
|                   | SRDF operations when R2 is target of TimeFinder Clone .....                   | 460        |
|                   | SRDF set operations for TimeFinder/Clone sessions .....                       | 463        |
|                   | SRDF set operations when R1 is source of TimeFinder/Clone .....               | 463        |
|                   | SRDF set operations when R1 is target of TimeFinder/Clone .....               | 463        |
|                   | SRDF set operations when R2 is source of TimeFinder/Clone .....               | 464        |
|                   | SRDF set operations when R2 is target of TimeFinder/Clone .....               | 464        |
|                   | SRDF operations for Extent-level TimeFinder/Clone sessions .....              | 466        |
|                   | SRDF operations when R1 is source of Extent-level Clone .....                 | 466        |
|                   | SRDF operations when R1 is target of Extent-level Clone .....                 | 468        |
|                   | SRDF operations when R2 is source of Extent-level Clone .....                 | 470        |
|                   | SRDF operations when R2 is target of Extent-level Clone .....                 | 473        |
|                   | SRDF set operations for Extent-level TimeFinder/Clone sessions .....          | 476        |
|                   | SRDF set operations when R1 is source of Extent-level Clone .....             | 476        |
|                   | SRDF set operations when R1 is target of Extent-level Clone .....             | 476        |
|                   | SRDF set operations when R2 is source of Extent-level Clone .....             | 477        |
|                   | SRDF set operations when R2 is target of Extent-level Clone .....             | 477        |
|                   | SRDF operations for TimeFinder Snapvx .....                                   | 478        |
|                   | Determine SnapVX states-SRDF operations interaction rules .....               | 478        |
|                   | SRDF operations when R1 is source of Snapvx .....                             | 478        |
|                   | SRDF operations when R1 is target for Snapvx .....                            | 480        |
|                   | SRDF operations when R2 is source of Snapvx .....                             | 482        |
|                   | SRDF operations when R2 is target for Snapvx .....                            | 484        |
|                   | SRDF set operations for TimeFinder Snapvx sessions .....                      | 487        |
|                   | SRDF set operations when R1 is source of TimeFinder Snapvx .....              | 487        |
|                   | SRDF set operations when R1 is target of TimeFinder Snapvx .....              | 487        |
|                   | SRDF set operations when R2 is source of TimeFinder Snapvx .....              | 488        |
|                   | SRDF set operations when R2 is target of TimeFinder Snapvx .....              | 488        |
| <b>Appendix C</b> | <b>SRDF operations and rcopy states</b>                                       | <b>489</b> |
|                   | rcopy session on the R1 side .....                                            | 490        |
|                   | Allowable SRDF operations when R1 is part of an rcopy PUSH .....              | 490        |

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| Allowable SRDF operations when R1 is part of an rcopy PULL..... | 492 |
| rcopy session on the R2 side.....                               | 494 |
| Allowable SRDF operations when R2 is part of an rcopy PUSH..... | 494 |
| Allowable SRDF operations when R2 is part of an rcopy PULL..... | 497 |



# FIGURES

|    |                                                                                |     |
|----|--------------------------------------------------------------------------------|-----|
| 1  | 2-site SRDF configurations.....                                                | 26  |
| 2  | SYMCLI command syntax.....                                                     | 30  |
| 3  | SRDF device and link states.....                                               | 45  |
| 4  | SRDF establish (full).....                                                     | 61  |
| 5  | SRDF establish (incremental).....                                              | 63  |
| 6  | Failback of an SRDF device.....                                                | 65  |
| 7  | Failover of an SRDF device.....                                                | 67  |
| 8  | Restore (full) an SRDF device.....                                             | 76  |
| 9  | Incremental restore an SRDF device.....                                        | 78  |
| 10 | Split an SRDF pair.....                                                        | 81  |
| 11 | Update SRDF device track tables.....                                           | 85  |
| 12 | SRDF/A legacy mode.....                                                        | 127 |
| 13 | SRDF/A multi-cycle mode.....                                                   | 128 |
| 14 | SRDF/Metro.....                                                                | 158 |
| 15 | SRDF/Metro Witness array and groups.....                                       | 160 |
| 16 | Setting up SRDF/Metro with Witness array; Before.....                          | 168 |
| 17 | Setting up SRDF/Metro with Witness array; After.....                           | 173 |
| 18 | Running redundant hosts to ensure consistency protection.....                  | 184 |
| 19 | Staging area for adding devices to the R1CG consistency group.....             | 200 |
| 20 | R1CG consistency group after a dynamic modify add operation.....               | 200 |
| 21 | Preparing the staging area for removing devices from the MyR1 CG.....          | 201 |
| 22 | MyR1 CG after a dynamic modify remove operation.....                           | 201 |
| 23 | Adding a device to independently-enabled SRDF groups of a concurrent CG.....   | 204 |
| 24 | Adding devices to independently-enabled SRDF groups of a cascaded CG.....      | 205 |
| 25 | Using an SRDF consistency group with a parallel database configuration.....    | 208 |
| 26 | Using an SRDF consistency group with BCVs at the target site.....              | 209 |
| 27 | Concurrent SRDF .....                                                          | 212 |
| 28 | Concurrent SRDF/S to both R2 devices.....                                      | 213 |
| 29 | Concurrent SRDF/A to both R2 devices.....                                      | 213 |
| 30 | Restoring the R1 a concurrent configuration.....                               | 216 |
| 31 | Restoring the source device and mirror in a concurrent SRDF configuration..... | 217 |
| 32 | Cascaded SRDF configuration.....                                               | 222 |
| 33 | Configuring the first hop.....                                                 | 226 |
| 34 | Configuring the second hop.....                                                | 227 |
| 35 | Determining SRDF pair state in cascaded configurations.....                    | 228 |
| 36 | Location of hop-2 devices.....                                                 | 229 |
| 37 | Cascaded SRDF with EDP.....                                                    | 230 |
| 38 | Set up first hop in cascaded SRDF with EDP.....                                | 232 |
| 39 | Set up second hop in cascaded SRDF with EDP.....                               | 232 |
| 40 | Adding a diskless SRDF mirror.....                                             | 235 |
| 41 | Cascaded configuration before planned failover.....                            | 236 |
| 42 | Planned failover - after first swap.....                                       | 237 |
| 43 | Planned failover - after second swap.....                                      | 237 |
| 44 | Cascaded SRDF/Star configuration.....                                          | 247 |
| 45 | Concurrent SRDF/Star configuration .....                                       | 248 |
| 46 | Typical concurrent SRDF/Star with R22 devices.....                             | 249 |
| 47 | Typical cascaded SRDF/Star with R22 devices.....                               | 249 |
| 48 | Site configuration transitions without concurrent devices.....                 | 253 |
| 49 | Site configuration transitions with concurrent devices.....                    | 253 |
| 50 | Concurrent SRDF/Star: normal operations.....                                   | 255 |
| 51 | Concurrent SRDF/Star: transient fault operations.....                          | 256 |

|     |                                                                              |     |
|-----|------------------------------------------------------------------------------|-----|
| 52  | Concurrent SRDF/Star: unplanned switch operations .....                      | 257 |
| 53  | Concurrent SRDF/Star: planned switch operations .....                        | 258 |
| 54  | Cascaded SRDF/Star: normal operations .....                                  | 259 |
| 55  | Cascaded SRDF/Star: transient fault operations (asynchronous loss) .....     | 260 |
| 56  | Cascaded SRDF/Star: transient fault operations (synchronous loss) .....      | 261 |
| 57  | Cascaded SRDF/Star: unplanned switch operations .....                        | 262 |
| 58  | Concurrent SRDF/Star setup using the StarGrp composite group .....           | 270 |
| 59  | Cascaded SRDF/Star setup using the StarGrp composite group .....             | 273 |
| 60  | Adding a device to a concurrent SRDF/Star CG .....                           | 290 |
| 61  | ConStarCG after a dynamic add operation .....                                | 291 |
| 62  | Adding devices to a cascaded SRDF/Star CG .....                              | 293 |
| 63  | CasStarCG after a dynamic add operation .....                                | 293 |
| 64  | Transient failure: concurrent SRDF/Star .....                                | 298 |
| 65  | Transient fault recovery: before reconfiguration .....                       | 300 |
| 66  | Transient fault recovery: after reconfiguration .....                        | 301 |
| 67  | Concurrent SRDF/Star: halted .....                                           | 303 |
| 68  | Concurrent SRDF/Star: switched .....                                         | 304 |
| 69  | Concurrent SRDF/Star: connected .....                                        | 305 |
| 70  | Concurrent SRDF/Star: protected .....                                        | 306 |
| 71  | Loss of workload site: concurrent SRDF/Star .....                            | 307 |
| 72  | Concurrent SRDF/Star: workload switched to synchronous site .....            | 309 |
| 73  | Concurrent SRDF/Star: new workload site connected to asynchronous site ..... | 310 |
| 74  | Concurrent SRDF/Star: protected to asynchronous site .....                   | 311 |
| 75  | Concurrent SRDF/Star: protect to all sites .....                             | 312 |
| 76  | Concurrent SRDF/Star: workload switched to asynchronous site .....           | 314 |
| 77  | Concurrent SRDF/Star: protected to asynchronous site .....                   | 315 |
| 78  | Concurrent SRDF/Star: one asynchronous site not protected .....              | 316 |
| 79  | Transient fault: cascaded SRDF/Star .....                                    | 318 |
| 80  | Cascaded SRDF/Star with transient fault .....                                | 319 |
| 81  | Cascaded SRDF/Star: asynchronous site not protected .....                    | 320 |
| 82  | SRDF/Star: after reconfiguration to concurrent .....                         | 322 |
| 83  | Cascaded SRDF/Star: halted .....                                             | 323 |
| 84  | Cascaded SRDF/Star: switched workload site .....                             | 324 |
| 85  | Loss of workload site: cascaded SRDF/Star .....                              | 325 |
| 86  | Workload switched to synchronous target site: cascaded SRDF/Star .....       | 326 |
| 87  | After workload switch to synchronous site: cascaded SRDF/Star .....          | 327 |
| 88  | Cascaded SRDF/Star after workload switch: protected .....                    | 328 |
| 89  | After reconfiguration to concurrent mode .....                               | 329 |
| 90  | Protected after reconfiguration from cascaded to concurrent mode .....       | 330 |
| 91  | Loss of workload site: Cascaded SRDF/Star .....                              | 331 |
| 92  | Cascaded SRDF: after switch to asynchronous site, connect, and protect ..... | 332 |
| 93  | Cascaded SRDF: after switch to asynchronous site .....                       | 333 |
| 94  | Halted cascaded SRDF/Star .....                                              | 335 |
| 95  | After reconfiguration to concurrent .....                                    | 336 |
| 96  | Halted cascaded SRDF/Star .....                                              | 337 |
| 97  | After reconfiguration to concurrent .....                                    | 338 |
| 98  | Halted cascaded SRDF/Star .....                                              | 339 |
| 99  | After cascaded path reconfiguration .....                                    | 340 |
| 100 | Halted concurrent SRDF/Star .....                                            | 341 |
| 101 | After reconfiguration to cascaded .....                                      | 342 |
| 102 | Halted concurrent SRDF/Star .....                                            | 343 |
| 103 | After reconfiguration to cascaded .....                                      | 344 |
| 104 | R1 migration: configuration setup .....                                      | 351 |
| 105 | R1 migration: establishing a concurrent relationship .....                   | 352 |
| 106 | R1 migration: replacing the source device .....                              | 353 |
| 107 | Migrating R2 devices .....                                                   | 354 |

|     |                                                                                     |     |
|-----|-------------------------------------------------------------------------------------|-----|
| 108 | R2 migration: configuration setup.....                                              | 355 |
| 109 | R2 migration: establishing a concurrent relationship.....                           | 356 |
| 110 | R2 migration: replacing the target device.....                                      | 357 |
| 111 | R1 migration example: Initial configuration.....                                    | 359 |
| 112 | Concurrent SRDF relationship.....                                                   | 362 |
| 113 | Migrated R1 devices.....                                                            | 364 |
| 114 | R2 migration example: Initial configuration.....                                    | 366 |
| 115 | Concurrent SRDF relationship.....                                                   | 367 |
| 116 | Migrated R2 devices.....                                                            | 368 |
| 117 | R1 migration: applicable R1/R2 pair states for migrate -setup.....                  | 369 |
| 118 | R2 migration: applicable R1/R2 pair states for migrate -setup.....                  | 370 |
| 119 | R1 migration: R11/R2 applicable pair states for migrate -replace (first leg).....   | 371 |
| 120 | R2 migration: R11/R2 applicable pair states for migrate -replace (first leg) .....  | 372 |
| 121 | R1 migration: applicable R11/R2 pair states for migrate -replace (second leg).....  | 373 |
| 122 | R2 migration: applicable R11/R2 pair states for migrate -replace (second leg) ..... | 374 |
| 123 | Automated data copy path in single-hop SRDF systems.....                            | 377 |
| 124 | Automated data copy path in multi-hop SRDF .....                                    | 381 |
| 125 | Concurrent BCV in a multi-hop configuration.....                                    | 384 |
| 126 | ECA consistent split .....                                                          | 398 |
| 127 | Commands used to perform splits in a complex configuration.....                     | 400 |
| 128 | Basic operations in multi-hop SRDF configurations.....                              | 402 |
| 129 | SnapVX and Cascaded SRDF.....                                                       | 404 |
| 130 | SnapVX and Concurrent SRDF.....                                                     | 406 |
| 131 | SRDF recovery environment.....                                                      | 408 |





# TABLES

|    |                                                                                |     |
|----|--------------------------------------------------------------------------------|-----|
| 1  | Revision history.....                                                          | 21  |
| 2  | SRDF documentation.....                                                        | 27  |
| 3  | SYMCLI SRDF commands .....                                                     | 31  |
| 4  | symrdf command options.....                                                    | 33  |
| 5  | Commands to display and verify SRDF, devices, and groups.....                  | 36  |
| 6  | Options for symrdf list command.....                                           | 40  |
| 7  | SRDF device and link states.....                                               | 45  |
| 8  | SRDF pair states.....                                                          | 46  |
| 9  | Possible SRDF device and link state combinations .....                         | 48  |
| 10 | Access rights required by an array.....                                        | 49  |
| 11 | SRDF control operations summary.....                                           | 54  |
| 12 | Device type combinations for creating SRDF pairs .....                         | 102 |
| 13 | Device pairs in storage groups.....                                            | 107 |
| 14 | SRDF device states before swap operation.....                                  | 119 |
| 15 | SRDF/A control operations .....                                                | 129 |
| 16 | Consistency modes for concurrent mirrors.....                                  | 192 |
| 17 | Allowable device types for adding devices to an RDF1 CG.....                   | 202 |
| 18 | Allowable device types for adding devices to a concurrent RDF1 CG.....         | 203 |
| 19 | Supported consistency modes for concurrent SRDF groups .....                   | 203 |
| 20 | Allowable device types for adding devices to a cascaded RDF1 CG.....           | 205 |
| 21 | Supported consistency modes for cascaded hops.....                             | 205 |
| 22 | Allowable device types for removing devices from an RDF1 CG .....              | 206 |
| 23 | Allowable device types for removing devices from a concurrent RDF1 CG.....     | 206 |
| 24 | Allowable device types for removing devices from a cascaded RDF1 CG.....       | 207 |
| 25 | SRDF modes for cascaded configurations (no EDP) .....                          | 223 |
| 26 | SRDF modes for cascaded configurations with EDP.....                           | 224 |
| 27 | SRDF modes allowed for SRDF/EDP .....                                          | 231 |
| 28 | SRDF/Star states.....                                                          | 251 |
| 29 | SRDF/Star target site states.....                                              | 251 |
| 30 | SRDF/Star operation categories.....                                            | 254 |
| 31 | SRDF/Star control operations.....                                              | 263 |
| 32 | symstar command options.....                                                   | 265 |
| 33 | Allowable SRDF/Star states for adding device pairs to a concurrent CG.....     | 292 |
| 34 | Allowable states for adding device pairs to a cascaded CG.....                 | 292 |
| 35 | Pair states of the SRDF devices after symstar modifycg -add completion.....    | 294 |
| 36 | Allowable states for removing device pairs from a concurrent SRDF/Star CG..... | 295 |
| 37 | Allowable states for removing device pairs from a cascaded SRDF/Star CG.....   | 295 |
| 38 | Possible pair states of the SRDF devices after a recovery.....                 | 297 |
| 39 | SRDF migrate -setup control operation and applicable pair states .....         | 368 |
| 40 | SRDF migrate -replace control operation and applicable pair states .....       | 370 |
| 41 | SRDF migrate -replace control operation and applicable pair states.....        | 372 |
| 42 | Initial setups for cycle timing parameters .....                               | 385 |
| 43 | Basic operations in a multi-hop configuration .....                            | 401 |
| 44 | symrecover options file parameters.....                                        | 413 |
| 45 | SRDF control operations and applicable pair states.....                        | 420 |
| 46 | R1 -> R21 cascaded SRDF control operations and applicable pair states.....     | 424 |
| 47 | R21 -> R2 cascaded SRDF control operations and applicable pair states.....     | 427 |
| 48 | R1 -> R21 cascaded SRDF set operations and applicable pair states.....         | 430 |
| 49 | R21 -> R2 Cascaded RDF Set Operations and Applicable Pair States.....          | 431 |
| 50 | SRDF control operations and applicable states for concurrent R1 pairs.....     | 432 |
| 51 | SRDF control operations and applicable states for concurrent R2 pairs.....     | 435 |

|    |                                                                                       |     |
|----|---------------------------------------------------------------------------------------|-----|
| 52 | SRDF control operations and applicable pair states for devices in an SRDF/CG.....     | 438 |
| 53 | Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap ....  | 440 |
| 54 | Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap ....  | 442 |
| 55 | Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap ....  | 445 |
| 56 | Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap ....  | 447 |
| 57 | Allowable SRDF set operations when R1 is the source of a TimeFinder/Snap.....         | 450 |
| 58 | Allowable SRDF set operations when R1 is the target of a TimeFinder/Snap.....         | 450 |
| 59 | Allowable SRDF set operations when R2 is the source of a TimeFinder/Snap.....         | 451 |
| 60 | Allowable SRDF set operations when R2 is the target of a TimeFinder/Snap.....         | 451 |
| 61 | Allowable SRDF operations when R1 is the source of a TimeFinder/Clone .....           | 453 |
| 62 | Allowable SRDF operations when R1 is the target of a TimeFinder/Clone .....           | 455 |
| 63 | Allowable SRDF operations when R2 is the source of a TimeFinder/Clone .....           | 458 |
| 64 | Allowable SRDF operations when R2 is the target of a TimeFinder/Clone .....           | 460 |
| 65 | Allowable SRDF set operations when R1 is the source of a TimeFinder/Clone.....        | 463 |
| 66 | Allowable SRDF set operations when R1 is the target of a TimeFinder/Clone.....        | 463 |
| 67 | Allowable SRDF set operations when R2 is the source of a TimeFinder/Clone.....        | 464 |
| 68 | Allowable SRDF set operations when R2 is the target of a TimeFinder/Clone.....        | 464 |
| 69 | Allowable SRDF operations when R1 is the source of an Extent-level Clone .....        | 466 |
| 70 | Allowable SRDF operations when the R1 is the target of an Extent-level Clone .....    | 468 |
| 71 | Allowable SRDF operations when R2 is the source of an Extent-level Clone .....        | 470 |
| 72 | Allowable SRDF operations when the R2 is the target of an Extent-level Clone .....    | 473 |
| 73 | Allowable SRDF set operations when R1 is the source of an Extent-level Clone.....     | 476 |
| 74 | Allowable SRDF set operations when R1 is the target of an Extent-level Clone.....     | 476 |
| 75 | Allowable SRDF set operations when the R2 is the source of an Extent-level Clone..... | 477 |
| 76 | Allowable SRDF set operations when R2 is the target of an Extent-level Clone.....     | 477 |
| 77 | SnapVX State Determination.....                                                       | 478 |
| 78 | Allowable SRDF operations when R1 is the source of a TimeFinder Snapvx.....           | 479 |
| 79 | Allowable SRDF operations when R1 is the target of a TimeFinder Snapvx.....           | 481 |
| 80 | Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx.....           | 483 |
| 81 | Allowable SRDF operations when R2 is the target of a TimeFinder Snapvx.....           | 485 |
| 82 | Allowable SRDF set operations when R1 is the source of a TimeFinder Snapvx.....       | 487 |
| 83 | Allowable SRDF set operations when R1 is the target of a TimeFinder Snapvx.....       | 487 |
| 84 | Allowable SRDF set operations when R2 is the source of a TimeFinder Snapvx.....       | 488 |
| 85 | Allowable SRDF set operations when R2 is the target of a TimeFinder Snapvx.....       | 488 |
| 86 | Allowed SRDF operations when the R1 is part of an rcopy PUSH .....                    | 490 |
| 87 | Allowed SRDF operations when the R1 is part of an rcopy PULL .....                    | 492 |
| 88 | Allowed SRDF operations when the R2 is part of an rcopy PUSH .....                    | 495 |
| 89 | Allowed SRDF operations when the R2 is part of an rcopy PULL .....                    | 497 |

# 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 technical support professional if a product does not function properly or does not function as described in this document.

---

## Note

This document was accurate at publication time. Go to EMC Online Support (<https://support.emc.com>) to ensure that you are using the latest version of this document.

---

## Purpose

## Audience

## Related documentation

The following EMC publications provide additional information:

- 
- 

## Special notice conventions used in this document

EMC uses the following conventions for special notices:

### **DANGER**

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

---

### **WARNING**

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

---

### **CAUTION**

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

---

### **NOTICE**

Addresses practices not related to personal injury.

---

## Note

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

---

## Typographical conventions

EMC uses the following type style conventions in this document:

|                         |                                                                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Bold</b>             | Used for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)                                          |
| <i>Italic</i>           | Used for full titles of publications referenced in text                                                                                                                                                                     |
| Monospace               | Used for: <ul style="list-style-type: none"> <li>• System code</li> <li>• System output, such as an error message or script</li> <li>• Pathnames, filenames, prompts, and syntax</li> <li>• Commands and options</li> </ul> |
| <i>Monospace italic</i> | Used for variables                                                                                                                                                                                                          |
| <b>Monospace bold</b>   | Used for user input                                                                                                                                                                                                         |
| []                      | Square brackets enclose optional values                                                                                                                                                                                     |
|                         | Vertical bar indicates alternate selections - the bar means “or”                                                                                                                                                            |
| { }                     | Braces enclose content that the user must specify, such as x or y or z                                                                                                                                                      |
| ...                     | Ellipses indicate nonessential information omitted from the example                                                                                                                                                         |

---

### Where to get help

EMC support, product, and licensing information can be obtained as follows:

#### Product information

For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at <https://support.emc.com>.

#### Technical support

Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

#### 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 [techpubcomments@emc.com](mailto:techpubcomments@emc.com).

# Revision history

The following table presents the revision history of this document:

**Table 1** Revision history

| Solutions Enabler | Description and/or change                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8.1               | <p>New content:</p> <ul style="list-style-type: none"> <li>• <a href="#">What's new in Solutions Enabler 8.1 on page 27</a></li> <li>• <a href="#">SRDF/Metro Operations on page 157</a></li> </ul> <p>Revised content:</p> <ul style="list-style-type: none"> <li>• <a href="#">Establish an SRDF pair (full) on page 60</a></li> <li>• <a href="#">Establish an SRDF pair (incremental) on page 62</a></li> <li>• <a href="#">Restore SRDF pairs (full) on page 75</a></li> <li>• <a href="#">Restore SRDF pairs (incremental) on page 77</a></li> <li>• <a href="#">Suspend I/O on links on page 83</a></li> <li>• <a href="#">Create an SRDF group and add pairs on page 92</a></li> <li>• <a href="#">Modifying dynamic SRDF groups on page 96</a></li> <li>• <a href="#">Removing dynamic SRDF groups on page 99</a></li> <li>• <a href="#">Create dynamic pairs with -file option on page 104</a></li> <li>• Suspend operation in <a href="#">Control operations for R1 - R2 pair states on page 420</a></li> <li>• Establish and full establish operations in <a href="#">SRDF operations when R1 is target of TimeFinder Clone on page 455</a></li> <li>• Establish and full establish operations in <a href="#">SRDF operations when R1 is target for Snapvx on page 480</a></li> <li>• New ActiveActive and ActiveBias states for operations in SRDF/Metro configurations: <a href="#">Control operations for R1 - R2 pair states on page 420</a></li> </ul> <p>Removed content:</p> <p>SRDF/A features and benefits are now published in <i>VMAX3 Family Product Guide</i>.</p> |
| 8.0.3             | <p>New restriction for the “update” operation when the R2 device is on an array running OS 5977.</p> <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 8.0.2             | <p>New content:</p> <ul style="list-style-type: none"> <li>• <a href="#">Cascaded Operations on page 221</a></li> <li>• <a href="#">SRDF/Star Operations on page 245</a></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

**Table 1** Revision history (continued)

| Solutions Enabler | Description and/or change                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>• <a href="#">SRDF operations for TimeFinder Snapvx on page 478</a></li> <li>• <a href="#">SRDF set operations for TimeFinder Snapvx sessions on page 487</a></li> <li>• <a href="#">TimeFinder SnapVX and SRDF on page 403</a></li> </ul> <p>Revised:</p> <ul style="list-style-type: none"> <li>• 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. New -sg option to manage devices by SG.<br/><a href="#">Create dynamic pairs with the -sg option on page 104</a></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 8.0.1             | <p>New content:</p> <ul style="list-style-type: none"> <li>• Track size increased to 128K: Geometry Compatibility Mode supports full SRDF functionality for devices of different track sizes.</li> <li>• Geometry Compatibility Mode</li> </ul> <p>Revised content:</p> <ul style="list-style-type: none"> <li>• SRDF/A's new multi-cycle mode creates multiple SRDF/A cycles on the R1 side at regular intervals, to provide smaller incremental updates to the R2. Creation of a new cycle is de-coupled from the data being committed to the R2 side. 2 fields in the output of the <code>symrdf query</code> command have new meaning in multi-cycle mode: <ul style="list-style-type: none"> <li>▪ Avg Transmit Cycle Time - Average time to transfer a Transmit Cycle from the R1 side to the R2 side.<br/>R1 Transmit cycles are transferred to the R2 side independent of the creation of a new Capture cycle on the R1 side.</li> <li>▪ Transmit Queue Depth on R1 Side - Total number of Transmit cycles on the R1 side.<br/>The total of R1 cycles is retrieved from the R1 side, regardless of which side (R1 or R2) the CLI command is executed on.</li> </ul> <p><i>EMC VMAX3 Family Product Guide</i> provides a detailed description of the changes to SRDF/A.</p></li> <li>• Multi-core, multi-ports per director: Specify both the director AND the ports for the SRDF emulation to use on each side. Modified commands include <code>symrdf addgrp</code> and <code>symrdf modifygrp</code>.</li> </ul> <p>Removed content (deprecated commands and options):</p> <ul style="list-style-type: none"> <li>• Adaptive copy write pending mode is not supported when the R1 side of the RDF pair is on an array running HYPERMAX OS.</li> <li>• HYPERMAX OS does not support meta devices. Device pairs between non-meta devices on an array running Solutions Enabler 8.0.1/HYPERMAX OS and a meta device on an array running Solutions Enabler 7.6/Enginuity 5876 are supported, but require the applicable Enginuity ePack on the array running Enginuity 5876.</li> <li>• <code>symrdf monitor</code> command, including all options and arguments. <code>symcfg monitor</code> command provides identical functionality.</li> <li>• -port option removed from the <code>symcfg -ra all list</code> command.</li> </ul> |

**Table 1** Revision history (continued)

| Solutions Enabler | Description and/or change                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>• -vxfs and -rdb options removed from the symreplicate start command.</li> <li>• SYMAPI_SYNC_DIRECTION option removed from the options file.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 7.6               | <p>New content:</p> <p>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 <i>VMAX RecoverPoint Splitter and Enhancements in Engenuity 5876</i></p> <p>Revised content:</p> <ul style="list-style-type: none"> <li>• SRDF operations and pair state tables</li> </ul> <p>Removed content:</p> <p>The following chapters were moved from this guide to the SolVe Desktop VMAX Family and DMX procedure generator:</p> <ul style="list-style-type: none"> <li>• Performing SRDF Control Operations</li> <li>• Querying and verifying with SRDF Commands</li> <li>• Implementing Consistency Protection</li> <li>• Performing SRDF/Automated Replication Operations</li> </ul> <p>The steps to download SolVe and the generator are described in “Related documentation”</p> |





# CHAPTER 1

## SRDF CLI overview

This chapter describes the following topics:

|                                                    |    |
|----------------------------------------------------|----|
| • <a href="#">Introduction to SRDF</a> .....       | 26 |
| • <a href="#">SYMCLI for SRDF</a> .....            | 30 |
| • <a href="#">SRDF pair states and links</a> ..... | 44 |
| • <a href="#">Before you begin</a> .....           | 48 |

## Introduction to SRDF

The EMC® Symmetrix® Remote Data Facility (SRDF®) family of products offers a range of array based disaster recovery, parallel processing, and data migration solutions for VMAX® Family systems, including:

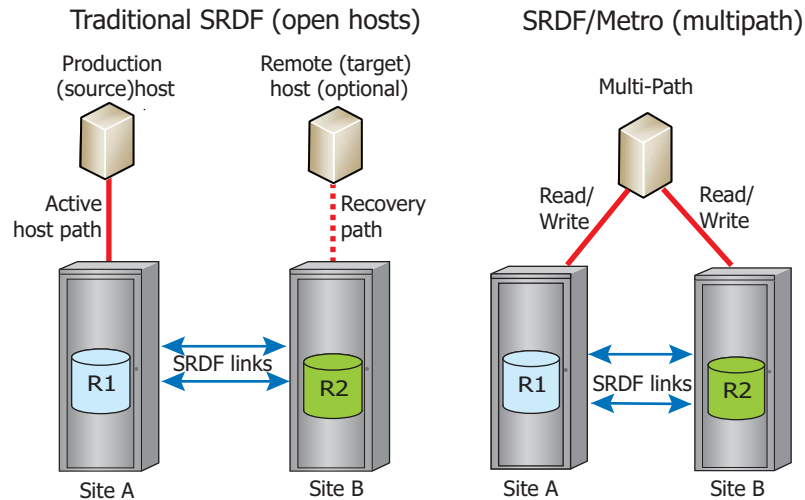
- HYPERMAX OS for VMAX 100K, 200K, and 400K arrays
- Engenuity for VMAX 10K, 20K, and 40K arrays

SRDF replicates data between 2, 3 or 4 arrays located in the same room, on the same campus, or thousands of kilometers apart. Replicated volumes may include a single device, all devices on a system, or thousands of volumes across multiple systems.

HYPERMAX OS Q3 2015 introduces a new SRDF configuration; SRDF/Metro.

The following image shows two-site SRDF configurations, one traditional and one SRDF/Metro.

**Figure 1** 2-site SRDF configurations



In *traditional* SRDF configurations:

- A host at the production site is connected to the local array.
- SRDF device pairs are designated as the R1 side (local to the host) and R2 side (remote)
- R1 and R2 device pairs are connected over SRDF links.
- The production host writes I/O to the R1 side of the device pair at the primary site.
- SRDF mirrors the production I/O to the R2 side of the device pair at the secondary site(s).

In *SRDF/Metro* configurations:

- R2 devices acquire the personality (geometry, device WWN) of the R1 device .
- R1 and R2 devices to appear to hosts(s) as a single virtual device across the two SRDF paired arrays
- The host (multiple hosts in clustered configurations) can read and write to both the R1 and R2 devices.
- For single host configurations, host I/Os are issued by a single host. Multi-pathing software directs parallel reads and writes to each array.

- For clustered host configurations, host I/Os can be issued by multiple hosts accessing both sides of the SRDF device pair.

## 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: <ul style="list-style-type: none"> <li>• SRDF Solutions</li> <li>• SRDF interfamily connectivity</li> <li>• SRDF concepts and terminology</li> <li>• SRDF/DM, SRDF/AR, SRDF/Concurrent</li> <li>• SRDF integration with other products</li> </ul> | <i>EMC VMAX3 Family Product Guide</i>                                                                                                                                                |
| 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 <i>SRDF 2-site Interfamily connectivity</i> or <i>SRDF 3-site Interfamily connectivity</i> . |
| Securing your configuration                                                                                                                                                                                                                                                                                                     | <i>EMC VMAX Family Security Configuration Guide</i>                                                                                                                                  |
| 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 -> <i>Managing SRDF using SYMCLI</i> .        |

## What's new in Solutions Enabler 8.1

This section describes changes introduced by the latest version of SRDF.

### SRDF/Metro

Starting with version 8.1, Solutions Enabler 8.1 supports SRDF/Metro. SRDF/Metro configurations are a significant change from traditional SRDF. In SRDF/Metro

configurations, R2 devices are Read/Write accessible to a host (or multiple hosts in clusters). Hosts write to both the R1 and R2 sides of SRDF device pairs, and SRDF/Metro ensures that each copy remains current and consistent.

Refer to [SRDF/Metro Operations on page 157](#).

SRDF/Metro features, functions and failover behaviors are described in *EMC VMAX3 Family with HYPERMAX Product Guide*

## SRDF restrictions

Solutions Enabler 7.6/Enginuity 5876 cannot discover arrays running Solutions Enabler 8.0.1/HYPERMAX OS or later.

## SRDF backward compatibility to Enginuity 5876

### SRDF/Metro

5876 arrays with the applicable ePack can participate only as Witness arrays in SRDF/Metro configurations.

Witness SRDF groups can be created between both of two VMAX3 arrays running HYPERMAX OS SR Q32015 or later and a 5876 array.

An SRDF/Metro configuration between the two VMAX3 arrays can then use Witness protection, provided by the 5876 array.

### Solutions Enabler 8.0.1

You can use SRDF features in Solutions Enabler 8.0.1/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 127](#).
- 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 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

Track size for FBA devices increased from 64K in Enginuity 5876 to 128K in HYPERMAX OS. Geometry Compatibility Mode supports full SRDF functionality for devices on arrays running Enginuity 5876 with an odd number of cylinders paired with devices on arrays running HYPERMAX OS.

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.

---

### Note

Devices that are part of an SRDF/Metro configuration cannot have the GCM attribute set.

---

### NOTICE

Do not set GCM on devices that are mounted and under Local Volume Manager (LVM) control.

---

- 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  $\frac{1}{2}$  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.

**NOTICE**

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`, `syndg show ld`, `syndg list ld -v`, `sympd show`, and `sympd list -v` commands have been enhanced to report the GCM attribute.

## SYMCLI for SRDF

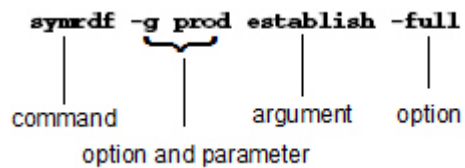
This section describes:

- [SYMCLI command syntax on page 30](#)
- [Get command help on page 30](#)
- [Set environmental variables on page 31](#)
- [Preset names and IDs on page 31](#)
- [Table 3 on page 31](#) lists the four main SRDF SYMCLI commands to establish, maintain and monitor SRDF configurations.
- [Table 4 on page 33](#) lists options for the `symrdf` command.
- [Table 5 on page 36](#) lists a variety of commands to display, query and verify your SRDF configuration.
- [Table 6 on page 40](#) 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.

**Figure 2** SYMCLI command syntax



## Get command help

### Description

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

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

### Examples

To display help for the `symrdf` command, enter:

```
symrdf - h
```

To display the man page for the `symrdf` command, enter:

```
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

## Set environmental variables

### Description

SYMCLI includes variables to streamline command line sessions.

### Examples

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

```
symcli -env
```

To view the variables that are set, enter:

```
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
```

## Preset names and IDs

### Description

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.

### Examples

To set the SID for all `-sid` arguments, enter:

```
set env SYMCLI_SID 000192601365
```

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

```
symcli -env
```

To view the current setting for all environment variables, enter:

```
symcli -def
```

## SYMCLI SRDF commands

**Table 3** SYMCLI SRDF commands

| Command | Description                                                                                                                                                                                                                                                               | For more information                                                                                                      |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| symrdf  | Control operations on SRDF devices, including: <ul style="list-style-type: none"> <li>• 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.</li> </ul> | See: <ul style="list-style-type: none"> <li>• <a href="#">Basic SRDF control operations summary on page 54</a></li> </ul> |

**Table 3** SYMCLI SRDF commands (continued)

| Command                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | For more information                                                                                                                                                     |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | <ul style="list-style-type: none"> <li>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.</li> <li>Splits an SRDF pair, which stops mirroring for the SRDF pairs in a device group.</li> <li>Fails over and back from the source (R1) side to the target (R2) side, switching data processing to the target (R2) side.</li> <li>Updates the source (R1) side after a failover, while the target (R2) side may still be operational to its local host(s).</li> <li>Swaps the source (R1) and target (R2) destinations between the target and the source.</li> <li>Creates, deletes, or swaps dynamic SRDF device pairs.</li> <li>Performs dynamic SRDF group controls to add, modify, and remove dynamic groups.</li> <li>Enables link domino locally or remotely when creating dynamic groups.</li> <li>Enables auto link recovery locally or remotely when creating dynamic groups.</li> <li>Enables/disables consistency for SRDF/A capable devices operating in asynchronous mode that are managed by a device group or file.</li> </ul> | <ul style="list-style-type: none"> <li><a href="#">Basic SRDF Control Operations on page 53</a></li> <li><code>symrdf man</code> page.</li> </ul>                        |
| <code>symreplicate</code> | <p>Invokes a replicate session that generates automated recurrent, background copies of the standard data following a path across SRDF links and cascading BCVs.</p> <p>Start, stop, and restart a replicate session.</p> <p>Used for SRDF/Automated Replication.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <p>See:</p> <ul style="list-style-type: none"> <li><a href="#">SRDF Automated Recovery Operations on page 407</a></li> <li><code>symreplicateman</code> page.</li> </ul> |
| <code>symstar</code>      | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>See:</p> <ul style="list-style-type: none"> <li><a href="#">SRDF/Star Operations on page 245</a></li> <li><code>symstar man</code> page.</li> </ul>                   |
| <code>symrecover</code>   | <p>Monitor the session state during attempts to restart a group session if it enters the suspended or partitioned state.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <p>See:</p> <ul style="list-style-type: none"> <li><a href="#">SRDF Automated Recovery Operations on page 407</a></li> <li><code>symrecover man</code> page.</li> </ul>  |



## 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

| 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 <code>list</code> , the -all option shows all SRDF mirrors of the selected devices. 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).<br><br><b>Note</b><br>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.                                                                                                                                                                                                  |
| -bias                 | Used with <code>suspend</code> to change the bias side of an SRDF/Metro configuration in addition to making the device pairs NR on the SRDF link.                                                                                                                                                                                                                                                              |
| -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 redispays, 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 <code>list</code> , 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 <i>Filename</i> | 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 <a href="#">Control operations for R1 - R2 pair states</a> .                                                                                             |
| -format               | Used with <code>createpair</code> 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.                                                                                                                                                                                                                                                 |

**Table 4** symrdf command options (continued)

| Option              | Description                                                                                                                                                                                                                                                                                                                                                   |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -full               | Requests a full establish or restore operation.                                                                                                                                                                                                                                                                                                               |
| -g <i>GroupName</i> | Specifies the device group for SRDF operations.                                                                                                                                                                                                                                                                                                               |
| -h                  | Provides brief, online help.                                                                                                                                                                                                                                                                                                                                  |
| -hop2               | For cascaded configurations, specifies a group's second-hop devices.                                                                                                                                                                                                                                                                                          |
| -hop2_rdfg          | Used with the createpair command that specifies a storage group. Specifies the SRDF group number at the second hop.<br>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.                                                                                                                                                                                                                                                                                 |
| -rdf_metro          | When used with the createpair action, indicates the SRDF pairs will be created in an SRDF/Metro configuration.                                                                                                                                                                                                                                                |
| -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.<br>When used -sg createpair -hop2, identifies the SRDF group associated with the specified storage group.<br><hr/> <b>Note</b><br><hr/> -hop2_rdfg specifies the SRDF group used to create the hop2 pair.<br><hr/>                                                                                                      |
| -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).                                                                                                                         |

**Table 4** symrdf command options (continued)

| Option       | Description                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | <hr/> <b>Note</b><br>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. <hr/>                                                                                                                                                                                                                                                                 |
| -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.                                                                                                                                                                                                                                                                                                                                                                      |
| -remote_sg   | Specifies the remote storage group name.<br>Used with createpair to specify the storage group.<br>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 <a href="#">Dynamic failover restore on page 122</a>                                                                                                                                                                                             |
| -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.<br><hr/> <b>Note</b><br>Use caution when applying this option as data can become lost or corrupted. <hr/> |
| -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 <a href="#">Write disable R1 on page 86</a> for more information.                                                                             |
| -use_bias    | When used with createpair -establish, createpair -restore, establish or restore actions, indicates that SRDF/Metro configuration will use bias instead of Witness protection.                                                                                                                                                                                                                                              |
| -v           | Provides more detailed, verbose command output.                                                                                                                                                                                                                                                                                                                                                                            |

**Table 4** symrdf command options (continued)

| Option                | Description                                                                                                                                                                                                                               |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-witness</code> | When used with <code>addgrp</code> , identifies the RDF group as a Witness SRDF group. When used with <code>removegrp</code> or <code>modifygrp</code> , specifies the action is targeted for an RDF group which is a Witness SRDF group. |

## Commands to display, query and verify SRDF configurations

The following table lists SYMCLI commands to display, query, and verify your SRDF configuration.

### Note

The following table is intended to provide examples of the types of information displayed by the `list` and `verify` commands. It is NOT a complete list of all options and states that can be verified. For a complete list, refer to the EMC Solutions Enabler CLI Command Reference

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

| SYMCLI command                                                                                                         | Description of command output                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>symcfg list</b>                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>symcfg list</code>                                                                                               | Displays the connectivity (Local or Remote) of each array.<br>Useful for verifying that only one array is connected to the host in a SRDF/Star configuration.                                                                                                                                                                                                                                                   |
| <code>symcfg list -v</code>                                                                                            | Displays a more detailed (verbose) listing, including: <ul style="list-style-type: none"> <li>Concurrent SRDF Configuration State</li> <li>Dynamic SRDF Configuration State</li> <li>Concurrent Dynamic SRDF Configuration</li> <li>RDF Data Mobility Configuration State</li> </ul>                                                                                                                            |
| <code>symcfg list -sid SID -rdfig {all  RDF_group_number}</code>                                                       | Displays SRDF group-level settings for a specific group or all groups on a array, such as: <ul style="list-style-type: none"> <li>Group type</li> <li>Director configuration</li> <li>Group flags, including auto link recovery, link domino, SRDF/Star mode, SRDF software and hardware compression, and SRDF single round trip</li> <li>SRDF flags, including consistency and SRDF status and mode</li> </ul> |
| <code>symcfg list -RA {all  Director }</code><br><code>symcfg list -RA {all  Director } -rdfig RDF_group_number</code> | Display all RDF directors, or a a specified RDF director.<br>Display RDF directors associated with a specified SRDF group.                                                                                                                                                                                                                                                                                      |
| <code>symcfg list -RA {all  Director } -p {all  Port}</code>                                                           | HYPERMAX OS only.<br>Display all ports or specified port for SRDF groups configured on all or the specified director: <ul style="list-style-type: none"> <li>Port ID</li> </ul>                                                                                                                                                                                                                                 |

**Table 5** Commands to display and verify SRDF, devices, and groups (continued)

| SYMCLI command                                              | Description of command output                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                             | <ul style="list-style-type: none"> <li>Negotiated speed (Gb/second)</li> <li>Maximum speed (Gb/second)</li> <li>Port status (online or offline)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>symcfg list -RA {all  Director } -p {all  Port</code> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>symdev list</b>                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symdev list -r1</code>                                | Displays only the R1 side of the SRDF configuration.<br>R1 devices not in a device group are displayed as N/Grp'd.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>symdev list -sid SID -r1 -bcv</code>                  | Displays the RDF1 BCV devices for the specified array.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>symdev list -sid SID -devs Device:Device -lock</code> | Display devices with a device external lock.<br>Displays a specified range of devices that have a device external lock.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>symdev show</b>                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symdev show Device_number -sid SID</code>             | Displays information about the specified SRDF devices, including: <ul style="list-style-type: none"> <li>SRDF device type and its group number</li> <li>Whether the device is in an SRDF/Metro configuration</li> <li>Whether the device is paired with a diskless or concurrent device</li> <li>Whether the device has a standard/thin relationship</li> <li>If the R2 device is larger than its R1</li> <li>Whether SRDF/A group-level and/or device-level write pacing is currently activated and supported for the SRDF/A session</li> <li>Whether the device is pace-capable</li> </ul> |
| <b>symdmg show</b>                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symdmg show group_name</code>                         | Displays detailed information about device groups, including RDF groups.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>symmir query</b>                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symmir -g RDF_group_name query</code>                 | Displays the BCV pairs in the specified device group and their state of mirroring.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>symrdf list</b>                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symrdf list</code>                                    | 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.<br>See <a href="#">Table 6 on page 40</a> for a list of symrdf list command options.                                                                                                                                                                                                                                                   |
| <b>symrdf query</b>                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>symrdf -g RDF_device_group_name query</code>          | Displays the state of the SRDF devices and their SRDF links in the specified device group.<br>During normal operations, the SRDF pair is Synchronized: <ul style="list-style-type: none"> <li>The R1 devices and SRDF links are read-writable.</li> </ul>                                                                                                                                                                                                                                                                                                                                    |

**Table 5** Commands to display and verify SRDF, devices, and groups (continued)

| SYMCLI command                                                         | Description of command output                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                        | <ul style="list-style-type: none"> <li>The R2 devices are write disabled.</li> <li>The link is in synchronous replication.<br/>During failed over operations:</li> <li>The R1 devices are write disabled.</li> <li>The R2 devices are read/write.</li> <li>The SRDF links are suspended.</li> </ul>                                                       |
| <code>symrdf -g RDF_device_group_name query -all</code>                | Displays the SRDF pair state of all devices in the specified device group, regardless of the device type.                                                                                                                                                                                                                                                 |
| <code>symrdf -g RDF_device_group_name query -bcv</code>                | Displays the SRDF pair state of the SRDF BCV devices in the specified device group.                                                                                                                                                                                                                                                                       |
| <code>symrdf -g RDF_device_group_name query -summary</code>            | Displays summarized information about the state of the SRDF devices and their SRDF links in the specified device group, including: <ul style="list-style-type: none"> <li>Pair state</li> <li>Number of invalid tracks on the source and target</li> <li>Synchronization rate</li> <li>Estimated time remaining for SRDF pair synchronization.</li> </ul> |
| <code>symrdf -cg composite_group_name query</code>                     | Displays the state of the SRDF devices and their SRDF links in the specified composite group.                                                                                                                                                                                                                                                             |
| <code>symrdf -sid SID -rdfg GrpNum -sg storage_group_name query</code> | Displays the state of the SRDF devices and their SRDF links in the specified storage group.                                                                                                                                                                                                                                                               |
| <b>symrdf verify (file)</b>                                            |                                                                                                                                                                                                                                                                                                                                                           |
| <code>symrdf -f Device_filename verify</code>                          | Verifies/displays the state of devices in the specified device file.                                                                                                                                                                                                                                                                                      |
| <code>symrdf -f Device_filename verify -activeactive</code>            | For SRDF/Metro configurations, verifies/displays whether any devices in the specified device file are in the 'ActiveActive' state.                                                                                                                                                                                                                        |
| <code>symrdf -f Device_filename verify -all -i 5 -synchronized</code>  | Verifies/displays a message every 5 seconds as to whether any devices in the specified device file are in the 'Synchronized' state until all SRDF pairs are synchronized.                                                                                                                                                                                 |
| <b>symrdf verify (group)</b>                                           |                                                                                                                                                                                                                                                                                                                                                           |
| <code>symrdf -g RDF_group_name verify</code>                           | Verifies/displays the state of devices in the specified device group.                                                                                                                                                                                                                                                                                     |
| <code>symrdf -g RDF_group_name verify -failedover</code>               | Verifies/displays whether any devices in the specified device group are in the 'Failed Over' state.                                                                                                                                                                                                                                                       |
| <code>symrdf -g RDF_group_name verify -synchronized</code>             | Verifies/displays whether any devices in the specified device group are in the 'Synchronized' state.                                                                                                                                                                                                                                                      |
| <code>symrdf -g RDF_group_name verify -i 30 -synchronized</code>       | Verifies/displays a message every 30 seconds as to whether any devices in the specified device group are in the 'Synchronized' state.                                                                                                                                                                                                                     |
| <code>symrdf -g RDF_group_name verify -all -i 5 -synchronized</code>   | 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.                                                                                                                                                                                |

**Table 5** Commands to display and verify SRDF, devices, and groups (continued)

| SYMCLI command                                                                                                          | Description of command output                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>symrdf -g <i>RDF_group_name</i> verify -split</code>                                                              | Verifies/displays whether any devices in the specified device group are in the 'Split' state.                                                                                                                                                                                                                                      |
| <code>symrdf -g <i>RDF_group_name</i> verify -syncinprog</code>                                                         | Verifies/displays whether any devices in the specified device group are in the 'SyncInProg' state.                                                                                                                                                                                                                                 |
| <code>symrdf -g <i>RDF_group_name</i> verify -activeactive</code>                                                       | For SRDF/Metro configurations, verifies/displays whether the SRDF device pairs are in the 'ActiveActive' state.                                                                                                                                                                                                                    |
| <code>symrdf -g <i>RDF_group_name</i> verify -activebias</code>                                                         | For SRDF/Metro configurations, verifies/displays whether the SRDF device pairs are in the 'ActiveBias' state.                                                                                                                                                                                                                      |
| <b>symrdf verify (composite group)</b>                                                                                  |                                                                                                                                                                                                                                                                                                                                    |
| <code>symrdf -cg <i>composite_group_name</i> verify</code>                                                              | Displays the state of devices in the specified composite group.                                                                                                                                                                                                                                                                    |
| <code>symrdf -cg <i>composite_group_name</i> verify -consistent</code>                                                  | Verifies/displays whether devices in the specified composite group are in the 'Consistent' state.                                                                                                                                                                                                                                  |
| <code>symrdf -cg <i>composite_group_name</i> verify -consistent -noinvalids -i 60</code>                                | 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. |
| <code>symrdf -cg <i>composite_group_name</i> verify -activeactive</code>                                                | For SRDF/Metro configurations, verifies/displays whether devices in the specified composite group are in the 'ActiveActive' state.                                                                                                                                                                                                 |
| <code>symrdf -cg <i>composite_group_name</i> verify -activebias</code>                                                  | For SRDF/Metro configurations, verifies/displays whether devices in the specified composite group are in the 'ActiveBias' state.                                                                                                                                                                                                   |
| <code>symrdf verify -summary -consistent -noinvalids -cg <i>composite_group_name</i> -i 45</code>                       | 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. |
| <b>symrdf verify (storage group)</b>                                                                                    |                                                                                                                                                                                                                                                                                                                                    |
| <code>symrdf -sg <i>storage_group_name</i> -sid <i>SID</i> -rdfg <i>RdfGrpNum</i> verify</code>                         | Verifies/displays the state of devices in the specified storage group.                                                                                                                                                                                                                                                             |
| <code>symrdf -sg <i>storage_group_name</i> -sid <i>SID</i> -rdfg <i>RdfGrpNum</i> verify -failedover</code>             | Verifies/displays whether any devices in the specified storage group are in the 'Failed Over' state.                                                                                                                                                                                                                               |
| <code>symrdf -sg <i>storage_group_name</i> -sid <i>SID</i> -rdfg <i>RdfGrpNum</i> verify -synchronized</code>           | Verifies/displays whether any devices in the specified storage group are in the 'Synchronized' state.                                                                                                                                                                                                                              |
| <code>symrdf -sg <i>storage_group_name</i> -sid <i>SID</i> -rdfg <i>RdfGrpNum</i> verify -i 30 -synchronized</code>     | Verifies/displays a message every 30 seconds as to whether any devices in the specified storage group are in the 'Synchronized' state.                                                                                                                                                                                             |
| <code>symrdf -sg <i>storage_group_name</i> -sid <i>SID</i> -rdfg <i>RdfGrpNum</i> verify -all -i 5 -synchronized</code> | 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.                                                                                                                                                        |

**Table 5** Commands to display and verify SRDF, devices, and groups (continued)

| SYMCLI command                                                                           | Description of command output                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -split</code>        | Verifies/displays whether any devices in the specified storage group are in the 'Split' state.                                                                                                                                                                                                                   |
| <code>symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -activeactive</code> | For SRDF/Metro configurations, verifies/displays whether devices in the storage group are in the 'ActiveActive' state.                                                                                                                                                                                           |
| <code>symrdf -sg storage_group_name -sid SID -rdfg RdfGrpNum verify -activebias</code>   | For SRDF/Metro configurations, verifies/displays whether devices in the storage group are in the 'ActiveBias' state.                                                                                                                                                                                             |
| <b>symstar list</b>                                                                      |                                                                                                                                                                                                                                                                                                                  |
| <code>symstar list</code>                                                                | Displays all the SRDF/Star composite groups visible to the host.                                                                                                                                                                                                                                                 |
| <code>symstar list -local</code>                                                         | Displays all the SRDF/Star composite groups local to your host.                                                                                                                                                                                                                                                  |
| <b>symstat command options</b>                                                           |                                                                                                                                                                                                                                                                                                                  |
| <code>-rdfg # ALL</code>                                                                 | Collect/display statistics for SRDF/A sessions by specified RA group number or all groups.                                                                                                                                                                                                                       |
| <code>-RepType rdf rdfa</code>                                                           | Collect/display statistics for the specified replication type.                                                                                                                                                                                                                                                   |
| <code>-type REQUESTS CACHE CYCLE RDF</code>                                              | Collect/display specified statistic type.<br>REQUEST (default) - I/O requests and throughput for device(s) and director(s).<br>CACHE - Cache activity for selected front-end or remote link director(s).<br>CYCLE - Active SRDF/A sessions or sessions that have non-zero cache usage.<br>RDF - SRDF/A sessions. |
| <code>-RE -RA ALL</code>                                                                 | Collect/display statistics for the specified SRDF director (-RA), GigE SRDF director (-RE) or both (ALL).                                                                                                                                                                                                        |
| <code>-rdflink</code>                                                                    | Collect/display SRDF link-level statistics.                                                                                                                                                                                                                                                                      |
| <code>-rdf_nw_comp</code>                                                                | Collect/display SRDF network compression (iSCSI statistics).                                                                                                                                                                                                                                                     |
| <code>-rdf_spdlmt</code>                                                                 | Collect/display SRDF speed limit information (iSCSI statistics).                                                                                                                                                                                                                                                 |

## symrdf list command options

The following table lists options for the `symrdf list` command, and describes the resulting output.

**Table 6** Options for `symrdf list` command

| symrdf list option | Description of output                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <code>-all</code>  | Lists all mirrors of the selected SRDF devices.                                                                                   |
| <code>-bcv</code>  | Lists only BCV devices.                                                                                                           |
| <code>-both</code> | Lists all SRDF devices that are RDF1 or RDF2 capable, when used with <code>-dynamic</code> .                                      |
| <code>-c</code>    | Specifies the number (count) of times to repeat the operation, displaying results appropriate to the operation at each iteration. |



**Table 6** Options for symrdf list command (continued)

| symrdf list option | Description of output                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -concurrent        | Lists concurrent SRDF (RDF11, RDF22, and RDF21) devices and the SRDF devices paired with a concurrent SRDF device.<br>When used with -R1, lists RDF11 devices and RDF1 devices that are paired with a concurrent SRDF device.<br>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.<br>To show the consistency state in the list of all the SRDF devices in array 333, enter:<br><code>symrdf -sid 333 -consistency list</code>                                                                                                                                                                                                            |
| -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.<br>When used with -R1, lists RDF1 devices that are either diskless or that are paired with a diskless device.<br>When used with -R2, lists RDF2 devices that are either diskless or are paired with a diskless device.<br>When used with -R21, lists RDF21 devices that are either diskless or that are paired with a diskless device. |
| -dup_pair          | Lists SRDF devices that are paired with the same SRDF type.<br>To list all of the duplicate pair devices in array 333, enter:<br><code>symrdf -sid 333 -dup_pair list</code><br><br><b>Note</b><br>Duplicate pair devices can result from an SRDF/Star failover scenario or a configuration change.                                                                                                                   |
| -dynamic           | Lists devices configured as dynamic SRDF.<br>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.<br>To list all of the half pair devices in array 333, enter:<br><code>symrdf -sid 333 -halfpair list</code><br><br><b>Note</b><br>Half pair devices can result from an SRDF/Star failover scenario, a <code>half_deletepair</code> operation, or a configuration change.                                                                                           |
| -nobcv             | Lists standard SRDF devices only (excludes SRDF BCV devices).                                                                                                                                                                                                                                                                                                                                                         |
| -R1<br>-R2<br>-R21 | Lists devices of RDF1 types (-R1), RDF2 types (-R2), or RDF21 types (-R21), respectively.                                                                                                                                                                                                                                                                                                                             |
| -rdf_metro         | List devices that are part of an SRDF/Metro configuration.                                                                                                                                                                                                                                                                                                                                                            |

**Table 6** Options for symrdf list command (continued)

| symrdf list option     | Description of output                                                                                                                                               |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -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, enter:<br><code>symrdf -sid 333 -resv list</code> |
| -star_mode             | Lists device that are SRDF/Star protected. For more information, refer to the <i>EMC VMAX3 Family Product Guide</i> .                                               |

## ping command

### Description

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

### Example

To ping SID 123, enter:

```
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*.

## verify command

### Description

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.

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.

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.

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 -rdg 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 -rdg 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 -rdg 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

## SRDF pair states and links

### NOTICE

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)
- Whether the device pair is part of an SRDF/Metro configuration, and
- The SRDF link state between the R1 and R2 devices

### Note

See [Invalid tracks in SRDF pairs](#) on page 47.

The following image shows states SRDF devices and links can report for SRDF/A, SRDF/S and SRDF/Metro configurations.

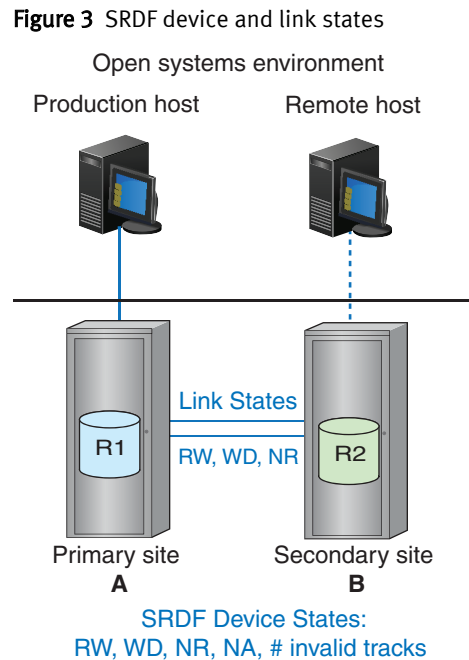


**Table 7** SRDF device and link states

|              |                                                                                                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                                                                                              |
| ActiveActive | <ul style="list-style-type: none"> <li>• R1 SRDF state is Ready.</li> <li>• SRDF link state is Ready.</li> <li>• R2 SRDF state is Ready.</li> <li>• R1 and R2 invalid tracks are 0.</li> </ul> |
| ActiveBias   | <ul style="list-style-type: none"> <li>• R1 SRDF state is Ready.</li> <li>• SRDF link state is Ready.</li> <li>• R2 SRDF state is Ready.</li> <li>• R1 and R2 invalid tracks are 0.</li> </ul> |

## SRDF pair states

An 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.

[Control operations for R1 - R2 pair states on page 420](#) lists control actions and the prerequisite SRDF pair state for each action, including:

- [Concurrent SRDF operations and applicable pair states on page 432](#)
- [Consistency group operations and applicable pair states on page 438](#)

- [Commands to display, query and verify SRDF configurations on page 36](#) describes the SYMCLI commands to verify pair states.

The following table lists the name and description of SRDF pair states.

**Table 8** SRDF pair states

| Pair State   | Description                                                                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SyncInProg   | A synchronization is currently in progress between the R1 and the R2 devices: <ul style="list-style-type: none"> <li>• There are existing invalid tracks between the two pairs, and</li> <li>• The logical links between both sides of an SRDF pair are up.</li> </ul>                                                                                                    |
| Synchronized | The R1 and the R2 are currently in a synchronized state: <ul style="list-style-type: none"> <li>• The same content exists on the R2 as the R1.</li> <li>• There are no invalid tracks between the two pairs.</li> </ul>                                                                                                                                                   |
| Split        | The R1 and the R2 are currently ready to their hosts, but: <ul style="list-style-type: none"> <li>• The links are not ready or,</li> <li>• Write disabled.</li> </ul>                                                                                                                                                                                                     |
| Failed Over  | The R1 is currently not ready or write disabled.<br>Operations have been failed over to the R2.                                                                                                                                                                                                                                                                           |
| R1 Updated   | The R1 is currently not ready or write disabled to the host: <ul style="list-style-type: none"> <li>• There are no local invalid tracks on the R1 side, and</li> <li>• The links are ready or write disabled.</li> </ul>                                                                                                                                                  |
| R1 UpdInProg | The R1 is currently not ready or write disabled to the host: <ul style="list-style-type: none"> <li>• There are invalid local (R1) tracks on the source side,</li> <li>• Data is being copied from the R2 to the R1 device, and the links are ready.</li> </ul>                                                                                                           |
| ActiveActive | The R1 and the R2 are currently in the default SRDF/Metro configuration which uses a Witness array: <ul style="list-style-type: none"> <li>• There are no invalid tracks between the two pairs.</li> <li>• The R1 and the R2 are Ready (RW) to the hosts.</li> </ul>                                                                                                      |
| ActiveBias   | The R1 and the R2 are currently in an SRDF/Metro configuration using bias: <ul style="list-style-type: none"> <li>• The user could have specified “use bias” during the establish/restore action or the Witness array is not available</li> <li>• There are no invalid tracks between the two pairs.</li> <li>• The R1 and the R2 are Ready (RW) to the hosts.</li> </ul> |
| Suspended    | The SRDF links have been suspended and are not ready or write disabled.<br>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.<br>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.                                   |

**Table 8** SRDF pair states (continued)

| Pair State    | Description                                                                                                                                                                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | A half pair and a duplicate pair are also reported as Partitioned.                                                                                                                                                                                                                                 |
| Mixed         | A composite SYMAPI device group SRDF pair state.<br>There are different SRDF pair states within a device group.                                                                                                                                                                                    |
| Invalid       | The default state when no other SRDF state applies. <ul style="list-style-type: none"> <li>The combination of the R1 device, the R2 device, and the SRDF link states do not match any other pair state.</li> <li>This state may occur if there is a problem at the disk director level.</li> </ul> |
| Consistent    | The R2 SRDF/A capable devices are in a consistent state.<br>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.

The following table lists:

- 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 9** 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 <sup>a</sup>          |
| R1 UpdInProg             | Not Ready or WD         | Ready (RW) or WD | Ready (RW)             | >0 <sup>a</sup>         |
| ActiveActive             | Ready (RW)              | Ready (RW)       | Ready (RW)             | 0                       |
| ActiveBias               | Ready (RW)              | Ready (RW)       | Ready (RW)             | 0                       |
| Split                    | Ready (RW)              | Not Ready or WD  | Ready (RW)             | —                       |
| SyncInProg               | Ready (RW)              | Ready (RW)       | Not Ready or WD        | >0                      |
| Suspended                | Any status <sup>b</sup> | Not Ready or WD  | Not Ready or WD        | —                       |
| Partitioned <sup>c</sup> | Any status              | Not Ready        | Not Available          | —                       |
| Partitioned <sup>d</sup> | Not Available           | Not Ready        | Any status             | —                       |
| Mixed                    | e                       | e                | e                      | —                       |
| Invalid <sup>e</sup>     | Any status <sup>f</sup> | Any status       | Any status             | —                       |
| Consistent               | Ready (RW) <sup>f</sup> | Ready (RW)       | Not Ready or WD        | 0                       |
| Transmit Idle            | Ready (RW) <sup>f</sup> | Ready (RW)       | Not Ready or WD        | —                       |

a. Refers to invalid local (R1) tracks on source.

b. Any status value is possible (Ready, Not Ready, Write Disabled, or Not Available).

c. Viewed from the host locally connected to the source (R1) device

d. Viewed from the host locally connected to the target (R2) device.

e. When no other SRDF states apply, the state defaults to Invalid.

f. The combination of source SRDF, SRDF links, and target SRDF statuses does 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](#) on page 49
- [Device external locks](#) on page 49
- [SRDF operations and copy sessions](#) on page 49
- [Mirror R1 to a larger R2 device](#) on page 49
- [Restrict synchronization](#) on page 50
- [SRDF software and hardware compression](#) on page 50
- [SRDF/A and the consistency exempt option](#) on page 51
- [Mixed-mode workloads on an SRDF director](#) on page 51



- [FAST VP SRDF coordination on page 52](#)

## Array access rights

Hosts must have specific access rights to an array to perform certain control operations. The following table lists common control operations and the required array access rights.

**Table 10** Access rights required by an array

| Operations                        | Required access rights |
|-----------------------------------|------------------------|
| <code>symrdf set rdfg</code>      | CFGSYM or SRDF         |
| <code>symrdf set rdfa</code>      | CFGSYM or SRDF         |
| <code>symrdf set rdfa_dse</code>  | CFGSYM or SRDF         |
| <code>symrdf set rdfa_pace</code> | CFGSYM or SRDF         |
| <code>symrdf addgrp</code>        | CFGSYM                 |
| <code>symrdf modifygrp</code>     | CFGSYM                 |
| <code>symrdf removegrp</code>     | CFGSYM                 |
| <code>symqos set IO</code>        | CFGSYM                 |
| <code>symqos reset IO</code>      | 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.

[Manage locked devices on page 394](#) 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 439](#) describes the TimeFinder/Snap and TimeFinder/Clone pair states, and which SRDF operations are available in each state.

## Mirror R1 to a larger R2 device

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

- SRDF/Metro configurations do not allow a larger R2 device.
- 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.

---

## Restrict synchronization

### Enginuity 5876 or lower

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

For sites running Enginuity 5876 or lower, the `SYMAPI_SYNC_DIRECTION` option in the options file allows you to confine SRDF and TimeFinder operations to only establish or restore actions.

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.

## HYPERMAX OS

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

## SRDF software and hardware compression

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

Both software and hardware compression can be activated 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.

Software and hardware compression can be enabled on both the R1 and R2 sides, but the actual compression happens from the side initiating the I/O.

Ensure that compression is enabled on the R1 side.

## Set compression for SRDF

### Syntax

To set hardware and software compression for an SRDF group, use the following form:

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

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

[Set SRDF group attributes on page 97](#) provides more information about SRDF group attributes.

### Options

#### on

Set the specified compression on.

#### off

Set the specified compression off.

### 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 list SRDF software and hardware compression status for all SRDF groups on SID 432:

```
symcfg list -rdfg all -sid 432
```

To list 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 or HYPERMAX OS 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.

## Mixed-mode workloads on an SRDF director

For arrays running Enginuity 5876 or later, 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*.

## Set mixed-mode workloads

### Syntax

Syntax for the symqos command:

```
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 HYPERMAX OS 5977, 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](#) ..... 54
- [SRDF basic control operations](#)..... 56

## Basic SRDF control operations summary

**Table 11** SRDF control operations summary

| Control operation                                                          | symrdf argument                                           | Description                                                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">SRDF modes of operation on page 56</a>                         | set mode<br>[sync asynch <br>acp_disk acp_wp <br>acp_off] | Set the replication mode for a device, device group, composite group, storage group, or list of devices in a device file.                                                                                                                                                             |
| <a href="#">Enable and disable SRDF consistency protection on page 188</a> | enable<br>disable                                         | Enables or disables consistency protection for SRDF/A capable devices.                                                                                                                                                                                                                |
| <a href="#">Establish an SRDF pair (full) on page 60</a>                   | establish -full                                           | Establishes remote mirroring and initiates a full data copy from the source (R1) device to the target (R2) device.<br>Use this for: <ul style="list-style-type: none"> <li>Initial synchronization of SRDF mirrors.</li> <li>Replacement of a failed drive on the R2 side.</li> </ul> |
| <a href="#">Establish an SRDF pair (incremental) on page 62</a>            | establish                                                 | Establishes remote mirroring and initiates an incremental data copy from the source (R1) device to the target (R2) device.<br>Use this to resynchronize after a split if you can discard the target data.                                                                             |
| <a href="#">Failback to source on page 64</a>                              | failback                                                  | Switches data processing from the target side (R2) back to the source (R1) side.<br>Use this to return the source site from the target site after resolving the cause of a failure.                                                                                                   |
| <a href="#">Failover to target on page 66</a>                              | failover                                                  | Switches data processing from the source (R1) side to the target (R2) side.<br>Use this when a failure occurs on the source side.                                                                                                                                                     |
| <a href="#">Invalidate R1 tracks on page 68</a>                            | invalidate r1                                             | Invalidates all tracks on the source (R1) side so that they can be copied over from the target (R2) side.                                                                                                                                                                             |
| <a href="#">Invalidate R2 tracks on page 68</a>                            | invalidate r2                                             | Invalidates all tracks on the target (R2) side so that they can be copied over from the source (R1) side.                                                                                                                                                                             |
| <a href="#">Make R1 ready on page 69</a>                                   | ready r1                                                  | Sets the source (R1) device to be SRDF ready to its local host.                                                                                                                                                                                                                       |
| <a href="#">Make R2 ready on page 70</a>                                   | ready r2                                                  | Sets the target (R2) device to be SRDF ready to its local host.                                                                                                                                                                                                                       |
| <a href="#">Make R1 not ready on page 70</a>                               | not_ready r1                                              | Sets the source (R1) device to be SRDF not ready to its local host.                                                                                                                                                                                                                   |
| <a href="#">Make R2 not ready on page 71</a>                               | not_ready r2                                              | Sets the target (R2) device to be SRDF not ready to its local host.                                                                                                                                                                                                                   |
| <a href="#">Merge track tables on page 71</a>                              | merge                                                     | Merges the track tables between the source (R1) and the target (R2) side.                                                                                                                                                                                                             |
| <a href="#">Move one-half of an SRDF pair on page 72</a>                   | half_movepair                                             | Moves one-half of the SRDF device pair to a different SRDF group.                                                                                                                                                                                                                     |

**Table 11** SRDF control operations summary (continued)

| Control operation                                                                         | symrdf argument | Description                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                           |                 | <b>Note</b><br>If the RA ends up supporting more than 64K devices in the new SRDF group, this operation fails.                                                                                                                                                                              |
| Move SRDF device pairs<br><a href="#">Move both sides of SRDF device pairs on page 72</a> | movepair        | Moves the SRDF device pair to a different SRDF group.<br><br><b>Note</b><br>If the RA ends up supporting more than 64K devices in the new SRDF group, this operation fails.                                                                                                                 |
| <a href="#">Read/write disable target device on page 73</a>                               | rw_disable r2   | Read/write disables the target (R2) device to its local host.                                                                                                                                                                                                                               |
| <a href="#">Refresh R1 on page 74</a>                                                     | refresh r1      | Marks any changed tracks on the source (R1) side to be refreshed from the R2 side.                                                                                                                                                                                                          |
| <a href="#">Refresh R2 on page 74</a>                                                     | refresh r2      | Marks any changed tracks on the target (R2) side to be refreshed from the R1 side.                                                                                                                                                                                                          |
| <a href="#">Restore SRDF pairs (full) on page 75</a>                                      | restore -full   | Resumes remote mirroring and initiates a full data copy from the target (R2) device to the source (R1) device.<br>Use this for: <ul style="list-style-type: none"> <li>Initial (reverse) synchronization of SRDF mirrors.</li> <li>Replacement of a failed drive on the R1 side.</li> </ul> |
| <a href="#">Restore SRDF pairs (incremental) on page 77</a>                               | restore         | Resumes remote mirroring and initiates an incremental data copy from the target (R2) device to the source (R1) device.<br>Use this for resynchronizing SRDF mirrors after a split if you can discard the source data.                                                                       |
| <a href="#">Resume I/O on links on page 79</a>                                            | resume          | Resumes I/O traffic on the SRDF links for the remotely mirrored SRDF pairs in the group.                                                                                                                                                                                                    |
| <a href="#">Split on page 80</a>                                                          | split           | Stops remote mirroring between the source (R1) device and the target (R2) device. The target device is made available for local host operations.<br>Use this when both sides require independent access, such as for testing purposes.                                                      |
| <a href="#">Suspend I/O on links on page 83</a>                                           | suspend         | Suspends I/O traffic on the SRDF links for the remotely mirrored SRDF pairs in the group.                                                                                                                                                                                                   |
| <a href="#">Swap SRDF pairs on page 84</a>                                                | 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.                                                                                                                                  |
| <a href="#">Swap one-half of an SRDF pair on page 84</a>                                  | 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.                                                                                                                       |
| <a href="#">Update R1 mirror on page 84</a>                                               | 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.                                                                                                                                                 |

**Table 11** SRDF control operations summary (continued)

| Control operation                           | symrdf argument  | Description                                                                                                                                           |
|---------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             |                  | 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. |
| <a href="#">Write disable R1 on page 86</a> | write_disable r1 | Write disables the source (R1) device to its local host.                                                                                              |
| <a href="#">Write disable R2 on page 87</a> | write_disable r2 | Write disables the target (R2) device to its local host.                                                                                              |
| <a href="#">Write enable R1 on page 87</a>  | rw_enable r1     | Write enables the source (R1) device to its local host.                                                                                               |
| <a href="#">Write enable R2 on page 88</a>  | 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.

### SRDF modes of operation

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.

#### Set the default SRDF 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.

#### SRDF/Metro Active mode

All device pairs in an SRDF/Metro configuration always operate in Active SRDF mode. Changes to or from Active mode are not allowed.

Writes can be done to both sides of the device pair. Data must be stored in cache at both sides before an acknowledgment is sent to the host that wrote the data.

#### Set the SRDF mode

##### Syntax

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

Refer to [symrdf createpair \(-file option\) syntax on page 102](#) for the `createpair` command syntax.

Use `symrdf set` command to set or modify the SRDF replication mode for a device group, a composite 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
```

### Options for *Mode*

#### **sync**

Sets the device pairs into synchronous mode.

#### **semi**

Sets the device pairs into semi-synchronous mode.

#### **acp\_disk**

Sets the device pairs to adaptive copy disk mode.

#### **acp\_wp**

Sets the device pairs to adaptive copy write pending mode.

Adaptive copy write pending mode is not supported when the R1 mirror of the RDF pair is on an array running HYPERMAX OS.

#### **acp\_off**

Turns off the adaptive copy mode for the device pairs.

#### **async**

Sets the device pairs to asynchronous mode.

## Set SRDF mode: synchronous

In the synchronous mode, the array responds to the host that issued a write operation to the source (R1) device only after the array containing the target (R2) device acknowledges that it has received and checked the data.

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

### Example

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

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

## Set SRDF 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. [Set adaptive copy disk skew on page 59](#) 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.

The following sections describe the commands to set the two types of adaptive copy mode:

- [Set SRDF mode: adaptive copy write pending on page 58](#)
- [Set SRDF mode: adaptive copy disk on page 58](#)

## Set SRDF 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/Engenuity 5876 backward compatibility**

In SRDF configurations where R1 devices are on an array running HYPERMAX OS, connected to one or more arrays are running Engenuity 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
```

## Set SRDF 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
```

**Set 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.

- 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
```

**Set SRDF 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.

---

**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 439](#) explains the applicable device pair states for TimeFinder/Snap or TimeFinder/Clone operations.

---

[SRDF/Asynchronous Operations on page 125](#) provides additional information.

**Example**

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

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

## Establish an SRDF pair (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.

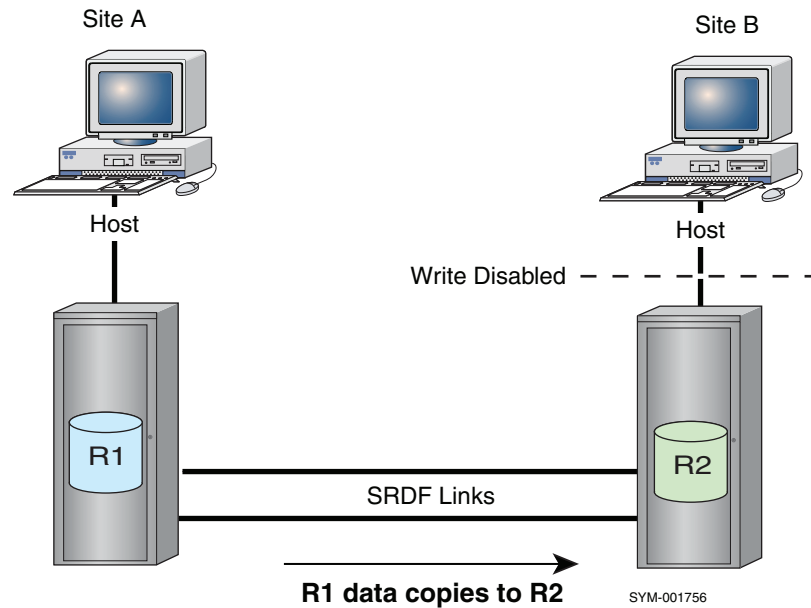
In SRDF/S configurations, when the establish control operation has successfully completed and the device pair is in the Synchronized state, the source (R1) device and the target (R2) device contain identical data.

In SRDF/A configurations, when the establish control operation has successfully completed and the device pair is in the Consistent state, the target (R2) device contains dependent write consistent data.

In SRDF/Metro configurations, once the source (R1) device and the target(R2) device contain identical data, the pair state is changed to either ActiveActive or ActiveBias and the R2 side is made RW-accessible to the host(s).

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.
- The following image shows establishing an SRDF pair.

**Figure 4** SRDF establish (full)**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 49 and [Table 5](#) on page 36.

[Control operations for R1 - R2 pair states](#) on page 420 lists the applicable SRDF pair states for this operation.

**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*.

**Syntax**

Use the `establish -full` operation for a device group, composite group, storage group, or device file:

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

Use the `-use_bias` option in SRDF/Metro configurations to indicate that the Witness method of determining bias is NOT used:

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

**NOTICE**

For SRDF/Metro configurations:

- The establish operation must include all devices in the group.
- If the Witness method is used to determine which side of the device pair remains accessible to the host, the Witness groups must be online.

---

[Create a device file on page 101](#) describes the steps to create a device file.

For SRDF/S configurations, use `verify` to confirm that the SRDF pairs are in the Synchronized state.

For SRDF/A configurations, use `verify` to confirm that the SRDF pairs are in the Consistent state.

For SRDF/Metro configurations, use `verify` to confirm that the SRDF pairs are in the ActiveActive or ActiveBias state.

**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
```

To establish all the pairs in an SRDF/Metro group using bias:

```
symrdf -f /tmp/device_file -sid 085 -rdfg 86 establish -full -use_bias
```

## Establish an SRDF pair (incremental)

An incremental establish re-synchronizes data on the source (R1) and target (R2) device when:

- A split RDF pair is rejoined.
- When device pairs are made Read-Write (RW) on the SRDF link after having been Not Ready (NR) on the link.

Only the new data that was updated on the source (R1) device while the SRDF pair was split or suspended 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.

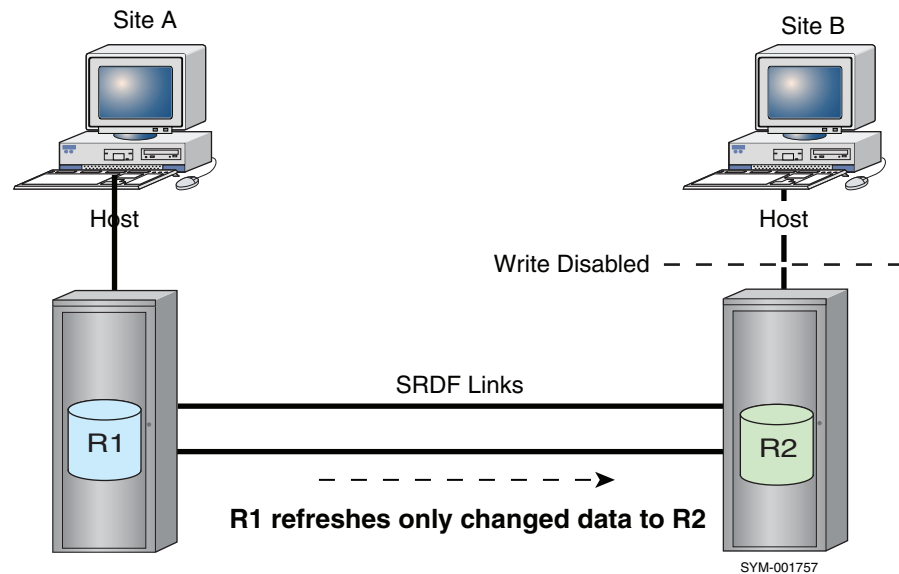
In SRDF/S configurations, when the establish control operation has successfully completed and the device pair is in the Synchronized state, the source (R1) device and the target (R2) device contain identical data.

In SRDF/A configurations, when the establish control operation has successfully completed and the device pair is in the Consistent state, the target (R2) device contains dependent write consistent data.

In SRDF/Metro configurations, once the source (R1) device and the target (R2) device contain identical data, the pair state is changed to either ActiveActive or ActiveBias and the R2 side is made RW-accessible to the host(s).

The following image shows an incremental establish of an SRDF pair.

**Figure 5** SRDF establish (incremental)



#### 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 49](#) and [Table 5 on page 36](#).

[Control operations for R1 - R2 pair states](#) lists the applicable SRDF pair states for this operation.

#### Syntax

#### Note

Incremental is the default for the establish operation. No option is required.

Issue the incremental -establish operation for a device group, composite group, storage group, or device file:

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

Use the `-use_bias` option in SRDF/Metro configurations to indicate that the Witness method of determining bias is NOT used:

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

#### NOTICE

For SRDF/Metro configurations:

- The establish operation must include all devices in the SRDF/Metro group.
- If the Witness method is used to determine which side of the device pair remains accessible to the host, the Witness groups must be online.

---

#### 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*

---

#### 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
```

To initiate an incremental establish for a list of SRDF pairs in SRDF/Metro group 86 where bias determines which side of the device pair remains accessible to the host:

```
symrdf -f /tmp/device_file -sid 085 -rdfg 86 establish -use_bias
```

## Failback to source

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.



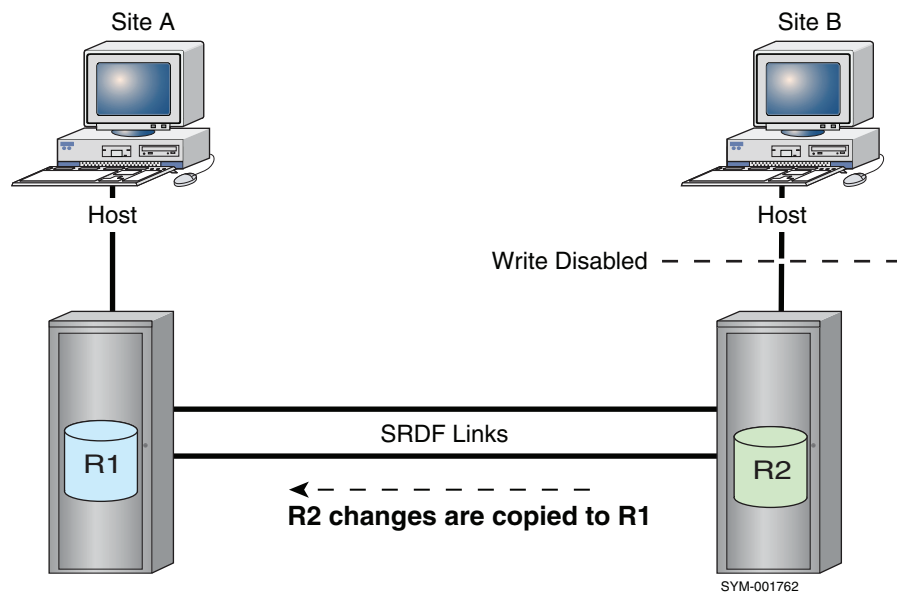
- The source (R1) device is read/write enabled to its local hosts.  
The target (R2) devices become read-only to their local hosts.

Failback includes the following general steps:

1. Stop I/Os on the failover host at site B.
2. Make all R2 devices in the 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.

The following image shows the failback of an SRDF pair.

**Figure 6** 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 49](#) and [Table 5 on page 36](#).

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

## Syntax

Issue the failback control operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName failback
symrdf -cg CgName failback
symrdf -cg SgName failback
symrdf -f[file] 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*

---

## 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 to target

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.

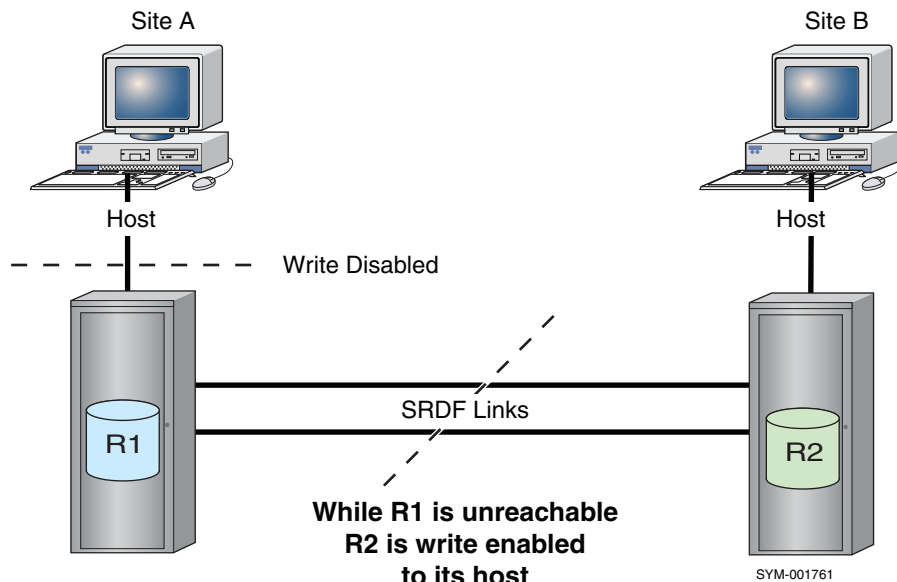
An *unplanned* failover moves production applications from the primary site to the secondary site after an unanticipated outage at the primary site, and the primary site is not available.

An 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.

The following image shows failover of an SRDF pair.

**Figure 7** Failover of an SRDF device



[Control operations for R1 - R2 pair states on page 420](#) 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 49](#) and [Table 5 on page 36](#).

#### Syntax

Issue the `failover` operation for a device group, composite group, storage group, or device file:

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

**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.

**Note**

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

**Syntax**

Issue the `invalidate r1` operation for a device group, composite group, storage group, or device file:

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

**Options****-nowd**

When used with the `invalidate r1` command, bypasses the validation check to ensure that the target of operation is write disabled to the host.

**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
```

**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.

**Note**

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

**Syntax**

Issue the `invalidate r2` operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName invalidate r2
symrdf -cg CgName invalidate r2
symrdf -sg SgName invalidate r2
symrdf -f[file] FileNameinvalidate r2
```

**Options****-nowd**

When used with the `invalidate r1` or `invalidate r2` commands, bypasses the validation check to ensure that the target of operation is write disabled to the host.

**Examples**

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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

**Syntax**

Issue the `ready r1` operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName ready r1
symrdf -cg CgName ready r1
symrdf -sg SgName ready r1
symrdf -f[file] FileNameready r1
```

**Examples**

To make the source (R1) device ready in all the SRDF 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

### Syntax

Perform the `not_ready r1` operation on a device group, composite group, storage group, or device file:

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

### 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

### Syntax

Issue the `ready r2` operation for a device group, composite group, storage group, or device file:

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

### 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for these operations.

### Syntax

Issue the `not_ready r2` operation for a device group, composite group, storage group, or 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
```

### Examples

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

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

To make the target (R2) device in one SRDF 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 SRDF 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, composite group, storage group, or device file. Use the merge operation to compare the track tables between SRDF device pairs that have been split and re-established.

### Syntax

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

Issue the `merge` operation for a device group, composite group, storage group, or device file:

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

**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 only one side of a dynamic SRDF pair from one SRDF group to another.

The current invalid track counters on both R1 and R2 are preserved, so resynchronization is required.

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

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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for control operations.

**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
```

**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 are preserved, so resynchronization is required.

**Note**

All devices that are moved together must have the same SRDF personality: from R1 to R1 or from R2 to R2.

**Syntax**

Move SRDF pairs using a device group, storage group, or device file:

```
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
```

[Move SRDF pairs on page 117](#) provides details on the `symrdf movepair` command for device files.



## Options

### **-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.

## Restrictions

The `movepair` operation has the following restrictions:

- The `-new_rdfg NewRDFgroup` argument and value are required.
- 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.
- The `-cons_exempt` option must be specified if the destination SRDF group supports an active SRDF/A session.
- The device pairs being moved must have been suspended using the `-cons_exempt` option if the original SRDF group supports an active SRDF/A session.

## Examples

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
```

The first device in each line of the device file moves to the new SRDF group. The second device in each line of the file moves to 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

## Syntax

Issue the `rw_disable r2` operation for a device group, composite group, storage group, or device file:

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

## 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

### Syntax

Issue the `refresh r1` operation for a device group, composite group, storage group, or 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

### Syntax

Issue the `refresh r1` and `refresh r2` operations for a device group, composite group, storage group, or 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 SRDF pairs (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.

---

### Note

Restore operations (incremental or full) are not allowed when the R2 device is larger than the R1 device.

---

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.

In SRDF/S configurations, when the restore 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.

In SRDF/A configurations, when the restore control operation has successfully completed and the device pair is in the Consistent state, the target (R2) device contains dependent write consistent data.

In SRDF/Metro configurations, once the source (R1) device and the target (R2) device contain identical data, the pair state is changed to either ActiveActive or ActiveBias and the R2 side is made RW-accessible to the host(s).

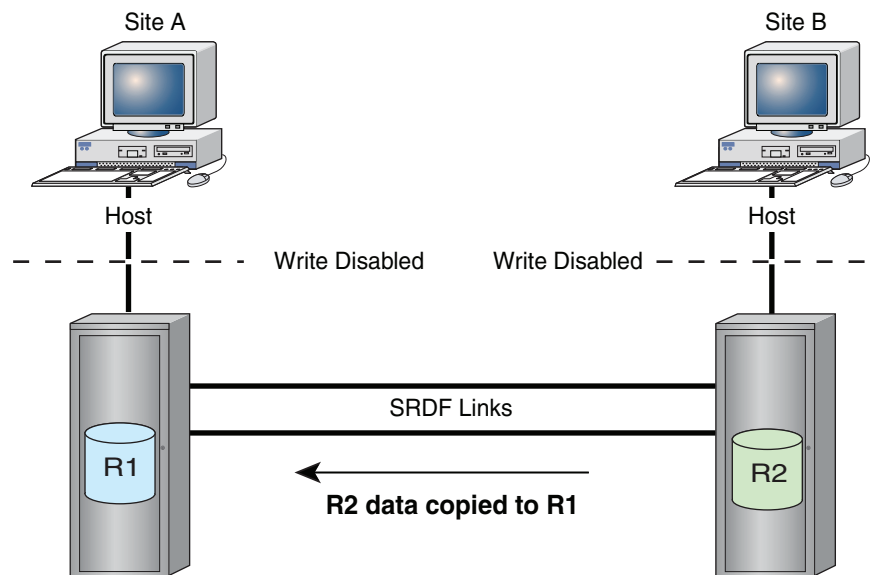
---

### 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

---

The following image shows restoring an SRDF pair.

**Figure 8** Restore (full) an SRDF device

[Control operations for R1 - R2 pair states on page 420](#) 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 49](#) and [Table 5 on page 36](#).

#### Syntax

Use the `restore -full` operation for a device group, composite group, storage group, or device file:

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

Use the `-use_bias` option in SRDF/Metro configurations to indicate that the Witness method of determining bias is NOT used:

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

For SRDF/A configurations, the restore operation must include all devices in the group unless the devices are `cons_exempt`.

For SRDF/Metro configurations:

- The restore operation must include all devices in the group.
- If the Witness method is used to determine which side of the device pair remains accessible to the host, the Witness groups must be online.

For SRDF/S configurations, use `verify` to confirm that the SRDF pairs are in the Synchronized state.

For SRDF/A configurations, use `verify` to confirm that the SRDF pairs are in the Consistent state.

For SRDF/Metro configurations, use `verify` to confirm that the SRDF pairs are in the ActiveActive or ActiveBias 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
```

To initiate a restore on a list devices in a SRDF/Metro group where bias determines which side of the device pair remains accessible to the host:

```
symrdf -f /tmp/device_file -sid 085 -rdfig 86 restore -full -use_bias
```

## Restore SRDF pairs (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.

---

### Note

Restore operations (incremental or full) are not allowed when the R2 device is larger than the 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.

In SRDF/S configurations, when the restore 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.

In SRDF/A configurations, when the restore control operation has successfully completed and the device pair is in the Consistent state, the target (R2) device contains dependent write consistent data.

In SRDF/Metro configurations, once the source (R1) device and the target (R2) device contain identical data, the pair state is changed to either ActiveActive or ActiveBias and the R2 side is made RW-accessible to the host(s).

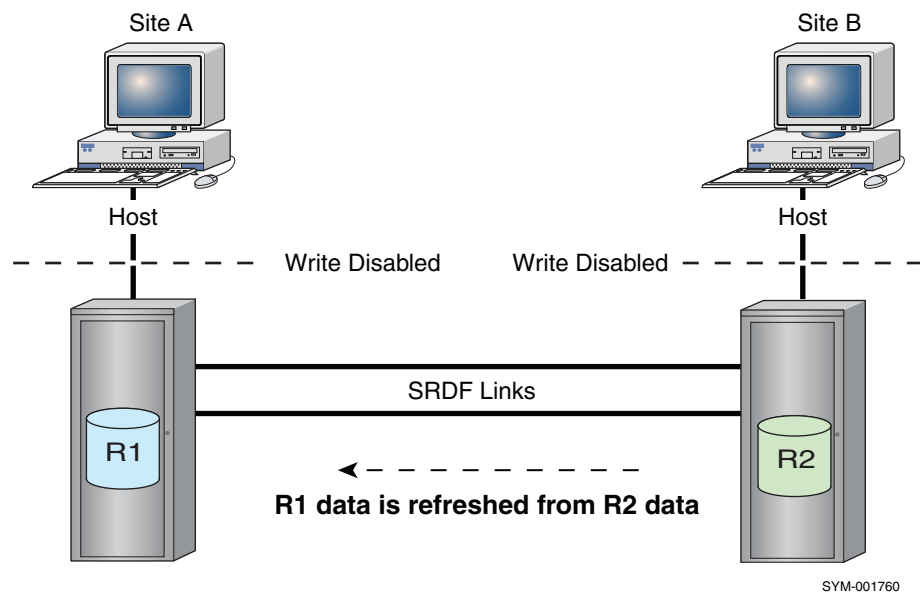
---

#### 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*

The following image shows the incremental restore of an SRDF pair.

**Figure 9** Incremental restore an SRDF device



[Control operations for R1 - R2 pair states on page 420](#) 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 49](#) and [Table 5 on page 36](#).

---

#### Syntax

---

#### Note

Incremental is the default for the restore operation. No option is required.

Issue the incremental restore operation for a device group, composite group, storage group, or device file:

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

Use the `-use_bias` option in SRDF/Metro configurations to indicate that the Witness method of determining bias is NOT used:

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

For SRDF/A configurations, the restore operation must include all devices in the group unless the devices are `cons_exempt`.

For SRDF/Metro configurations:

- The restore operation must include all devices in the group.
- If the Witness method is used to determine which side of the device pair remains accessible to the host, the Witness groups must be online.

For SRDF/S configurations, use `verify` to confirm that the SRDF pairs are in the Synchronized state.

For SRDF/A configurations, use `verify` to confirm that the SRDF pairs are in the Consistent state.

For SRDF/Metro configurations, use `verify` to confirm that the SRDF pairs are in the ActiveActive or ActiveBias 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
```

To initiate an incremental restore on a list devices in a SRDF/Metro group where bias determines which side of the device pair remains accessible to the host:

```
symrdf -f /tmp/device_file -sid 085 -rdfg 86 restore -use_bias
```

## Resume I/O on links

The `resume` operation resumes I/O traffic on the SRDF links.

For storage groups and device files, the operation applies to all SRDF pairs in the group or file.

For device groups and composite groups, the operation can be applied to all or only selected members of the group.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

**Syntax**

Issue the `resume` control operation for a device group, composite group, storage group, or 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 storage group `prod_sg`:

```
symrdf -sg prod_sg 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 only the specified SRDF pairs in device group `prod`:

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

## Split

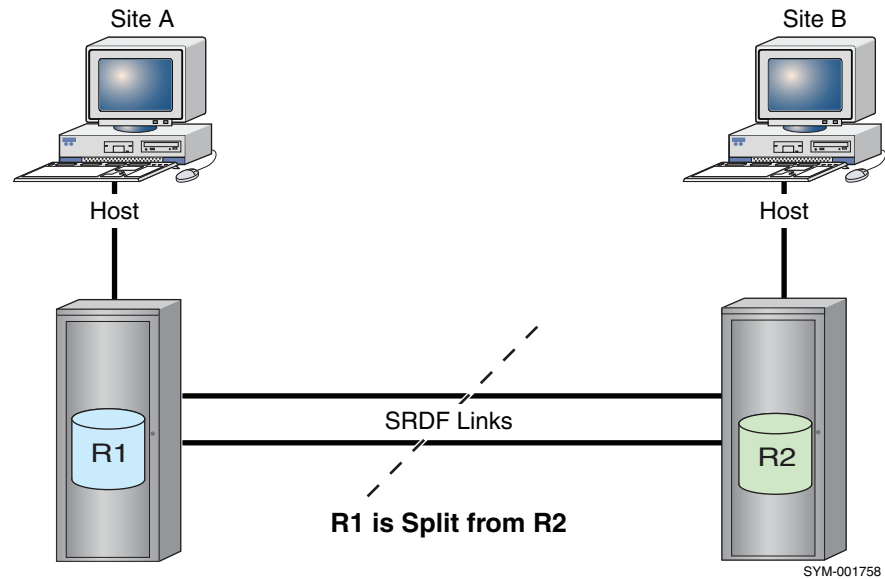
Split SRDF pairs when you require read and write access to the target (R2) side of one or more devices in a device group, composite group, storage group, or device file.

A split performs the following activities for each specified SRDF pair:

- 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.

The following image shows splitting an SRDF pair.



**Figure 10** Split an SRDF pair

[Control operations for R1 - R2 pair states on page 420](#) 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 49](#) and [Table 5 on page 36](#).

---

#### Syntax

Issue the `split` operation for a device group, composite group, storage group, or device file:

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

---

#### Note

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

---

#### 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

---

### Note

See also: [Consistency Group Operations on page 181](#)

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 `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.

## Suspend I/O on links

The `suspend` action suspends I/O traffic on the SRDF links for the specified 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.

---

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for this operation.

### 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.

---

### Syntax

Use the `suspend` operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName suspend [-immediate | -cons_exempt][-bias R1|R2]
symrdf -cg CgName suspend [-immediate | -cons_exempt][-bias R1|R2]
symrdf -sg SgName suspend [-immediate | -cons_exempt][-bias R1|R2]
symrdf -f[ile] FileName suspend [-immediate | -cons_exempt][-bias R1|R2]
```

### Options

#### -immediate

For SRDF/A configurations, causes the suspend command to drop the SRDF/A session immediately.

#### -cons\_exempt

Allows devices to be suspended without affecting the state of the SRDF/A session or requiring that other devices in the session be suspended.

#### -bias R1|R2

For SRDF/Metro configurations, specifies which side will be the bias side.

### 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
```

## 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.

### Restrictions

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
```

## 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.

---

### Note

The current states of the various devices involved in the SRDF swap must be considered before executing a swap action. [Table 14 on page 119](#) lists which states are legal for this operation.

---

### 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.

### 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
```

## 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.

---

#### Note

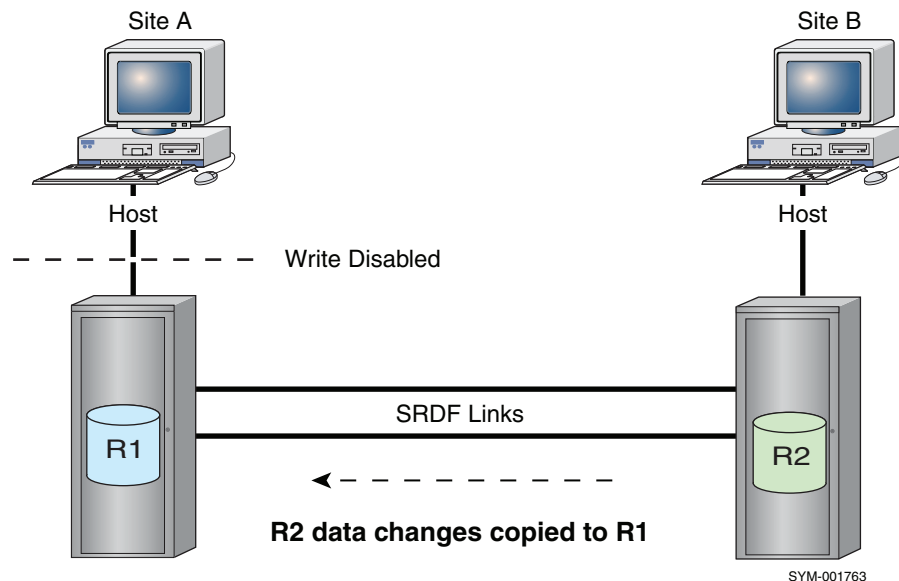
If you perform an update while the SRDF pair is Suspended and not ready at the source, the SRDF pair types are in an Invalid state as the update completes. To resolve this condition, use the `rw_enable r1` command to make the SRDF pairs become Synchronized.

---

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

The following image shows an update of an SRDF pair.

**Figure 11** Update SRDF device track tables



[Control operations for R1 - R2 pair states on page 420](#) 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 49](#) and [Table 5 on page 36](#).

---

**Syntax**

Issue the `update` operation for a device group, composite group, storage group, or device file:

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

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.

**Options****-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 triggered. The update sequence loops until the invalid track count is less than the number specified by the `#` value. If the invalid track count is less than the number of tracks specified by the `-until #` value, the command exits, otherwise, the following sequence of operations for update R1 mirror is retrigged until the threshold is reached.

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

**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
```

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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for these operations.

**Syntax**

Issue the `write_disable r1` operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName write_disable r1
```

```
symrdf -cg CgName write_disable r1
symrdf -sg SgName write_disable r1
symrdf -f[file] 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for these operations.

### Syntax

Issue the `write_disable r2` operation for a device group, composite group, storage group, or device file:

```
symrdf -g DgName write_disable r2
symrdf -cg CgName write_disable r2
symrdf -sg SgName write_disable r2
symrdf -f[file] 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for these operations.

### Syntax

Issue the `rw_enable r1` operation for a device group, composite group, or device file:

```
symrdf -g DgName rw_enable r1
symrdf -cg CgName rw_enable r1
symrdf -f[file] 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for these operations.

**Syntax**

Issue the `rw_enable r2` operation for a device group, composite group, or device file:

```
symrdf -g DgName rw_enable r2  
symrdf -cg CgName rw_enable r2  
symrdf -f[file] 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:

- [Dynamic operations overview](#) ..... 90
- [Manage SRDF groups](#) ..... 92
- [Device pairing operations](#) ..... 101
- [Group, move and swap dynamic devices](#) ..... 116

## Dynamic operations 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.

## SRDF group maximum per OS

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 and SRDF groups

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](#) on page 91
- [Domino mode](#) on page 91
- [Autolink recovery](#) on page 92
- [Hardware compression](#) on page 92
- [Software compression](#) on page 92

---

### Note

SRDF/A device groups have additional configurable attributes. See [Set SRDF/A group cycle time, priority, and transmit idle](#) on page 132.

---

## 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 Engenuity 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. [Manage transmit idle](#) on page 147 provides more information.

### NOTICE

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 mode 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.

# Manage SRDF groups

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

- [Create an SRDF group and add pairs on page 92](#)
- [Set SRDF group attributes on page 97](#)
- [Add/remove supporting directors for an SRDF group on page 98](#)
- [Removing dynamic SRDF groups on page 99](#)

## Create an SRDF group and add pairs

### SRDF/Metro

HYPERMAX OS SR Q32015/Solutions Enabler 8.1 introduces SRDF/Metro.

SRDF/Metro is a significant departure from traditional SRDF.

In SRDF/Metro configurations, R2 devices on VMAX3 arrays can be Read/Write accessible to hosts. SRDF/Metro R2 devices acquire the federated personality of the primary R1 device (geometry, device WWN). This federated personality of the R2 device causes the R1 and R2 devices to appear to hosts(s) as a single virtual device across the two SRDF paired arrays.

By default, SRDF/Metro configurations use "Witness" groups to determine which side of the SRDF device pair remains R/W accessible to the host in the event of link or other failures.

[Witness SRDF groups on page 161](#) provides more information.

### Multi-cores, multi-ports per director

In 5876 and earlier, all front-end emulations supported a fixed number (up to two) of ports. Multiple front-end emulations could exist on the same director board, providing additional host connectivity, but all such front-end directors were limited to one or two physical ports.

VMAX 100K/200K/400K arrays (referred to as VMAX3 arrays) running HYPERMAX OS/ Solutions Enabler 8.0.1 support a single front-end emulation of each type (FA, EF, etc.) per director, but each of these emulations supports a variable number of physical ports. Both the SRDF Gigabit Ethernet (RE) and SRDF Fibre Channel (RF) emulations can use any port on the director. The relationship between the SRDF emulation and resources on a director is configurable:

- 1 director/1 or multiple CPU cores/1 or multiple ports

Connectivity is not bound to a fixed number of CPU cores. You can change the amount of connectivity without changing CPU power.

The SRDF emulation supports up to 16 front-end ports per director (4 front-end modules per director), any or all of which can be used by SRDF. Both the SRDF Gigabit Ethernet and SRDF Fibre Channel emulations can use any port.

---

### Note

If hardware compression is enabled, the maximum number of ports per director is 12.

When you create an SRDF group on VMAX 100/200/400 arrays, you must select both the director AND the ports for the SRDF emulation to use on each side.

[Create an SRDF group and add pairs on page 92](#) and [Add/remove supporting directors for an SRDF group on page 98](#) have been modified to specify multiple ports.

A new option `-port <#|ALL>` for the `symsan list -sanrdf` command displays either the specified port number (0 - 31) or all ports. Output of the command displays fields for local and remote ports and status for local port.

### Syntax

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

```
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
  -witness
```

### Required options

`-sid SID`

Specifies the ID of the array where the group is added.

`-label GrpLabel`

Specifies a label for a dynamic SRDF group.

`-rdfg GrpNum`

Specifies an SRDF group number. Valid values are 1 - 250.

`-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.

**-remote\_rdfg** *GrpNum*

Specifies the SRDF group number on the remote array.

**-remote\_sid** *SID*

Specifies the ID of the remote array.

### Optional options

**-fibre | -gige | -farpoint**

Specifies the communication protocol for the group: Fibre Channel, Gigabit Ethernet, or FarPoint.

**-link\_domino {on|off}**

Enables/disables [Establish an SRDF pair \(full\) on page 60](#).

**-remote\_link\_domino {on|off}**

enables [Establish an SRDF pair \(full\) on page 60](#) remotely.

**-auto\_link\_recovery {on|off}**

Enables/disables local [Autolink recovery on page 92](#).

**-remote\_auto\_link\_recovery**

Enables [Autolink recovery on page 92](#) remotely.

**-link\_limbo** *0 - 120*

Enables [Link limbo on page 91](#).

**-rem\_link\_limbo** *0 - 120*

Enables [Link limbo on page 91](#) remotely.

**-witness**

Identifies the SRDF group as a Witness group.

### Restrictions

The following are requirements for adding a dynamic 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.

### Example - 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
```

**Example - 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
```

**Example - 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
```

**Creating a dynamic SRDF group****Procedure**

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.

**Note**

The two SRDF group numbers can be the same or different. Valid values are 1 - 250.

- Assigns a group label (-label dynggrp4) 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.
- 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 dynggrp4 -dir 12a
-remote_rdfg 4 -remote_sid 6240 -remote_dir 13a
```

**NOTICE**

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
```

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

## Modifying dynamic SRDF groups

Use the `symrdf set rdfg` command to set the attributes for an existing SRDF group, including:

- Link limbo
- Domino mode



- Autolink recovery
- Hardware compression
- Software compression

Use the `symrdf modifygrp` command to modify an existing SRDF group, including:

- Ports on a local director
- Ports on a remote director

Use the `-witness` option to modify Witness groups in SRDF/Metro configurations.

### Syntax

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

```
symrdf modifygrp {-add | -remove}
                 -rdfg GrpNum|-label GrpLabel
                 -sid SID
.....
                 -dir Dir:Port,Dir:Port,...
                 -remote_dir Dir:Port,Dir:Port,...
                 -witness
```

### Options

**-dir *Dir:Port, Dir:Port***

Specifies a comma-separated list of one or more local director:port combinations 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.

**-witness**

Identifies the group as an SRDF/Metro Witness group.

## Set SRDF group attributes

---

### Note

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

---

### Syntax

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

```
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]
```

### Options

**-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](#) on page 91.

-domino {on|off}  
Enables/disables [Domino mode](#) on page 91

-autolink\_recovery {on|off}  
Enables/disables [Autolink recovery](#) on page 92

-hwcomp {on|off}  
Enables/disables [Hardware compression](#) on page 92

-swcomp {on|off}  
Enables/disables [Software compression](#) on page 92

---

#### 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 Symmetrix Array Controls CLI Product 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
```

## Add/remove supporting directors for an SRDF group

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.

#### NOTICE

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.

**Example - Modify a group using 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
```

**Example - Modify a group using 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
```

**Example - Modify a group in a mixed configuration**

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
```

**Example - Remove a director**

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

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

## Removing dynamic SRDF groups

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.
- At least one physical connection between the local and remote array must exist.
- In SRDF/Metro configurations:
  - The Witness group cannot be removed if an SRDF/Metro group is currently using that Witness group for protection.
  - The Witness group can be removed if it is protecting an SRDF/Metro configuration(s) if there is another Witness available to provide the protection. The Witness group can be removed and the new Witness array will start protecting the SRDF/Metro group(s).

---

### Note

Deleting the group removes all local and remote director support.

---

**Syntax**

Use the `symrdf deletepair` command to remove all devices from the group.

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

```
symrdf removegrp -sid SymmID
                  -rdfg GrpNum | -label GrpLabel
                  -noprompt
```

```
-i Interval
-c Count
-star
-symforce
-witness
```

### Options

**-remote\_rdfg *GrpNum* -label *GrpLabel***

Specifies the SRDF group number on the remote array.

**-noprompt**

Prompts are not displayed after the command is entered.

**-i *Interval***

Interval, in seconds, between attempts to acquire an exclusive lock on the array host database or on the local and/or remote arrays.

**-c *Count***

Number (count) of times to attempt to acquire an exclusive lock on the array host database, or on the local and/or remote arrays.

**-star**

Specifies the action is targeted for devices in STAR mode.

**-symforce**

Requests the array force the operation to be executed when normally it is rejected.

#### NOTICE

When used with `removegrp`, removes one side of a dynamic SRDF group if the other side is not defined or is not accessible.  
Do not use this option except in emergencies.

---

**-witness**

Identifies the SRDF group as a Witness group.

### Example - Remove an SRDF group

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 dynggrp4
```

## Remove an SRDF group from one side of an SRDF configuration

### Restrictions

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.

### Syntax

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

### Example

The following example removes `dynggrp4` from array 180 on the local side:

```
symrdf removegrp -sid 180 -label dynggrp4 -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.

### Create a device file

To create a device file to define SRDF device pairs:

1. Create a text file containing two columns
2. 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.

#### NOTICE

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/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.

#### Note

The maximum symdev number that can be used on the HYPERMAX OS array is FFBF (65471).

#### 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.

When the `symrdf createpair -file FileName` command processes the device file, the `-type` option determines whether the devices in the left column are R1 or R2.

```
vi RDFG148
0060 0092
0061 0093
0062 0094
0063 0095
0064 0096
0065 0097
0066 0098
```

### Supported device types for SRDF pairs

The following table lists the valid device type combinations for creating an SRDF pair.

**Table 12** Device type combinations for creating SRDF pairs

| Device 1          | Device 2                |
|-------------------|-------------------------|
| Standard          | Standard                |
| Thin              | Thin                    |
| Standard          | Diskless <sup>a</sup>   |
| Thin <sup>b</sup> | Diskless <sup>a,b</sup> |
| Thin <sup>c</sup> | Standard <sup>d</sup>   |

- a. 5876 diskless devices cannot be paired with devices on HYPERMAX OS.
- b. FBA devices require Enginuity 5875 or higher. CKD devices are not supported.
- c. FBA devices require Enginuity 5875 or higher. CKD devices require Enginuity 5876 Q42012 SR or higher.
- d. Only on Enginuity versions 5671, 5773.50154, or 5875 and higher.

## Block createpair when R2 is larger than R1

### NOTICE

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 is larger than its R1 is blocked.

## Creating SRDF device pairs

This section explains the various ways to create dynamic SRDF device pairs.

### symrdf createpair (-file option) syntax

Use the `createpair` command to create SRDF device pairs.

```
symrdf -file Filename -sid SymmID
  -rdfg GrpNum
  -bypass
  -noprompt
  -i Interval
  -c Count
  -v
  -noecho
  -force
  -symforce
  -star

createpair
  -type <R1|R2>
  -remote_sg <SgName>
  -invalidate R1|R2 | -establish | -restore [-rp] |format -establish]>
  -hop2_rdfg <GrpNum>]
  -rdf_mode sync | semi | acp_wp | acp_disk | async
  -remote
  -nowd
```

---

**Note**

[Create device pairs in SRDF/Metro configurations on page 162](#) describes creating SRDF device pairs in SRDF/Metro configurations.

---

**Options**

**-file *Filename***

Specifies a device file for SRDF operations.

**-rdg *GrpNum***

Targets the operation to a specific SRDF group.

When used with `-sg createpair -hop2`, identifies the SRDF group associated with the SG.

**-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.

**-remote\_sg**

When used with `-hop2_rdg GrpNum`, the remote storage group for the second-hop.

**-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 once the SRDF pairs are created.

**-restore**

Begins copying data to the source devices, synchronizing the dynamic SRDF pairs once the SRDF pairs are created.

**-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 [Block createpair when R2 is larger than R1 on page 102](#).

**-g *GrpName***

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.

**-hop2\_rdfg**

Specifies the SRDF group number for the second-hop. Applicable only for `createpair -hop2` for an SG.

**-nowd**

Bypasses the check explained in [Verify host cannot write to target devices with -nowd option on page 112](#).

**Example**

In the following example:

- `-file` indicates devices are created using a device file *devices*.
- `-g ProdDB` indicates devices are created in device group *ProdDB*.
- `-sid` indicates local source array is SID 810.
- `-invalidate -r2` indicates that the R2 devices will be refreshed from the R1 source devices.
- `-type RDF1` indicates devices listed in the left column of the device file are configured as the R1 side.

```
symrdf createpair -g ProdDB -file devices -sid 810 -rdfg 2 -invalidate  
r2 -nop -type RDF1
```

## Create dynamic pairs with -file option

[Create a device file on page 101](#) 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
```

## Create dynamic pairs with the -sg option

Starting in Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR you can manage SRDF operations using 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. *EMC Solutions Enabler Array Management CLI User Guide* provides more information about storage groups.

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 GrpNum` 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.

This section describes the following topics:

- [Pair devices using storage groups on page 106](#)
- [Pair mixed devices using storage groups on page 108](#)
- [Pair devices in cascaded storage groups on page 108](#)
- [Pair devices in storage groups \(second hop\) on page 109](#)

**symrdf createpair (-sg option) syntax**

Use the `createpair` command with the `-sg` option to create SRDF device pairs using storage groups.

```
symrdf -sg SgName -sid SymmID -rdfg GrpNum
    -bypass
    -noprompt
    -i Interval
    -c Count
    -v | -noecho | -force | -symforce | -star
    -hop2

createpair
    -type <R1|R2>
    -remote_sg SgName
    -invalidate R1|R2 | -establish | -restore [-rp]
    -format | -establish
    -hop2_rdfg GrpNum]
    -rdf_mode sync | semi | acp_wp | acp_disk
    -remote
    -cons_exempt
    -nowd
```

**Options****-sg SgName**

Specifies a storage group for SRDF operations.

**-rdfg GrpNum**

Targets the operation to a specific SRDF group.

When used with `-sg createpair -hop2`, identifies the SRDF group associated with the SG.

**-type [R1|R2]**

Specifies whether the devices are configured as the R1 side or the R2 side.

**-remote\_sg SgName**

When used with `-hop2_rdfg GrpNum`, the remote storage group for the second-hop.

**-invalidate [R1|R2]**

Marks the source (R1) devices or the target (R2) devices to invalidate for a full copy when an SRDF pair is created.

**-establish**

Begins copying data to invalidated targets, synchronizing the dynamic SRDF pairs once the SRDF pairs are created.

**-restore**

Begins copying data to the source devices, synchronizing the dynamic SRDF pairs once the SRDF pairs are created.

**-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.

**-hop2\_rdfg *GrpNum***

Specifies the SRDF group number for the second-hop. Applicable only for `createpair -hop2` for an SG.

**-rdf\_mode *Mode***

Sets the SRDF mode of the pairs to be one of the following:

- synchronous (sync),
- 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 SRDF 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 [Block createpair when R2 is larger than R1 on page 102](#).

**-remote**

Requests a remote data copy. When the link is ready, data is copied to the SRDF mirror.

**-nowd**

Bypasses the check explained in [Verify host cannot write to target devices with -nowd option on page 112](#).

## Pair 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. Geometry Compatible Mode (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 on page 29](#) 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.

**NOTICE**

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<br>Dev | Pdev Name | Device<br>Config | Sts | Cap<br>(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<br>Dev | Device<br>Pdev Name | Cap<br>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 250 -sg localSG -type r1 -remote_sg remoteSG
```

After the operation, pairings are as follows:

**Table 13** Device pairs in storage groups

| 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                | 3278 MB     |

**Table 13** Device pairs in storage groups (continued)

### Pair 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<br>Dev | Pdev Name | Device<br>Config  | Sts | Cap<br>(MB) |
|------------|-----------|-------------------|-----|-------------|
| 000A0      | N/A       | <b>RDF1</b> +TDEV | RW  | 3278        |
| 000A1      | N/A       | TDEV              | RW  | 1875        |
| 000B1      | N/A       | <b>RDF2</b> +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 250 -sg localSG -type r1 -remote_sg remoteSG
```

After the operation, device A0 is an R11 and device B1 is an R21:

| Sym<br>Dev | Pdev Name | Device<br>Config   | Sts | Cap<br>(MB) |
|------------|-----------|--------------------|-----|-------------|
| 000A0      | N/A       | <b>RDF11</b> +TDEV | RW  | 3278        |
| 000A1      | N/A       | TDEV               | RW  | 1875        |
| 000B1      | N/A       | <b>RDF21</b> +TDEV | RW  | 4125        |
| 000C1      | N/A       | TDEV               | RW  | 3278        |

### Pair 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.

### Examples

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
```

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
```

### Pair 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.

#### NOTICE

To create pairs using the -hop2 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 Dev      | Pdev Name | Device Config | Sts | Cap (MB) |
|--------------|-----------|---------------|-----|----------|
| <b>01AA0</b> | N/A       | RDF1+TDEV     | RW  | 3278     |
| <b>01AB1</b> | N/A       | RDF1+TDEV     | RW  | 4125     |

paired with 2 R2 devices in storage group SG\_ABC on remote SID 086 (hop 1):

| Logical Device | Sym Dev | T E | R1 Inv Tracks | R2 Inv Tracks | K S | Sym Dev      | T... E... |
|----------------|---------|-----|---------------|---------------|-----|--------------|-----------|
| N/A            | 01AA0   | RW  | 0             | 0             | NR  | <b>0007A</b> | WD...     |
| N/A            | 01AB1   | RW  | 0             | 0             | NR  | <b>0007B</b> | 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 Dev | Pdev Name | Device Config     | Sts | Cap (MB) |
|---------|-----------|-------------------|-----|----------|
| 0009A   | N/A       | <b>RDF2</b> +TDEV | RW  | 3278     |
| 0009B   | N/A       | <b>RDF2</b> +TDEV | RW  | 4125     |

The devices at hop 1 that were R2 before the operation, are now R21 devices.

## Create pairs with the -establish option

### Note

In traditional SRDF configurations, 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.

### 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
```

## Create pairs with the -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.

### Restrictions

The `symrdf createpair -format` option has the following restrictions:

- Engenuity 5876 and higher.
- Not supported in concurrent SRDF configurations.
- Not supported in SRDF/Metro configurations.
- SRDF device pairs cannot be created in an SRDF Witness group
- 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 -rdfig 117  
-rdf_mode sync -nop
```

## Create pairs with the -invalidate option

### Syntax

Use the `symrdf createpair` command with the `-invalidate r1` or `-invalidate r2` option to create devices (R1 or R2) in a new or existing configuration.

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.
- Use the `restore` command to start copying to the invalidated source device.

```
symrdf -sid 55 -file devicefile establish -rdfig 1
```

Use the `query` command to check the progress of the establish operation:

```
symrdf -sid 55 -file devicefile query -rdfig 1
```

Once synchronized, you can perform various SRDF operations on SRDF pairs listed in the device file.

### 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 r2` 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 -rdfig 1 -type R1 -invalidate r2 -nowd
```

## Create pairs with the -restore option

Use the `-restore` option to copy data back to the R1 source devices.

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.

### Restrictions

- The device cannot be the source or target of a TimeFinder/Snap operation.

- Devices cannot be in the backend not ready state.
- Not allowed if an optimizer swap is in progress on a device.
- The emulation type must be same (such as, AS/400 has specific pairing rules).
- Not allowed if there are local invalid tracks on either the local or remote device.
- Not allowed if an SRDF/A session is active and `-cons_exempt` is not specified.
- Not allowed if the SRDF group is in asynchronous mode and the devices being added are not the same SRDF type R1 or R2.
- Not allowed if the SRDF group is in asynchronous mode with the SRDF links suspended and the `-restore` option is selected.
- Not allowed if the SRDF group is enabled for SRDF consistency protection.
- Not allowed if 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.
- SRDF device pairs cannot be created in an SRDF Witness group

#### Example

```
symrdf createpair -sid 55 -file devicefile -rdfg 1 -type R1 -restore
```

### Verify 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`

## Create 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 Engenuity 5875 or higher, both mirrors of a concurrent R1 device can be operating in SRDF/A mode.

[Concurrent Operations on page 211](#) provides more information.

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.

#### Restrictions

The following restrictions apply to creating dynamic concurrent SRDF pairs:



- 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.

### Examples

In a previous example, the `createpair` command created dynamic device pairs in 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
```

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

Use the `createpair` command with the `-restore` `-remote` options to restore the standard R2 type devices.

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 -rdfg 2 -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.

## Delete 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.

---

### 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.

### Restrictions

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.

### NOTICE

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
```

## Clear local invalid tracks

For Engenuity 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.

---

### Note

This functionality is not supported for diskless devices and does not delete any device pairs containing R11, R21, or R22 devices.

---

### Examples

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
```

## Delete one-half of an SRDF pair

The `half_deletepair` command dynamically removes 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.

---

### Note

In Concurrent SRDF configurations, the concurrent SRDF device is converted to a non-concurrent SRDF 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). Devices in Concurrent SRDF configurations are converted to non-concurrent SRDF devices.

[Cascaded SRDF control operations and applicable pair states on page 424](#) lists the applicable SRDF pair states for `half_deletepair` operations.

---

### 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 configuration changes.

---

### Restrictions

The `symrdf half_deletepair` command 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
```

## Group, move and swap dynamic devices

This section describes the steps to group, move and swap dynamic SRDF devices.

### Creating a device group using a device file

Device groups are the primary method to manage SRDF devices.

An SRDF device file allows you to manage the devices specified in the file as a single entity.

To create a device group:

#### Procedure

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 perform operations on all the dynamic SRDF pairs in the group.

For example, establish the group:

```
symrdf -g NewGrp establish
```

## Move dynamic SRDF device pairs

Prior to Engenuity 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 Engenuity 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.

## Move 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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for `movepair` operations.

### 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.

---

### Restrictions

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.

## Move one-half of an SRDF pair

The `half_movepair` command moves only one side of a dynamic SRDF pair from one SRDF group to another.

The current invalid track counters on both R1 and R2 are preserved, so resynchronization is required.

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

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.

[Control operations for R1 - R2 pair states on page 420](#) lists the applicable SRDF pair states for control operations.

### 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
```

## 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

The following general steps are required to perform an R1/R2 personality swap and resume SRDF operations:

1. Suspend the SRDF remote mirroring.
2. Perform a personality swap by converting the R1 to R2 and the R2 to R1 devices.
3. Determine the synchronization direction and synchronize the R1 and the R2 devices.
4. 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.

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.

## 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.

The following table lists which states are legal for this operation.

**Table 14** 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        |

## Display SRDF swap-capable devices

### Syntax

Use the `symrdf list` command with the `-dynamic` option to display SRDF devices configured as dynamic SRDF-capable:

```
symrdf list -dynamic [-R1] [-R2] [-both]
```

### Options

Use the command with no options to display all SRDF-capable devices.

#### -R1

Display all dynamic SRDF-capable devices that are configured as capable of becoming R1.

#### -R2

Display all dynamic SRDF-capable devices that are configured as capable of becoming R2.

#### -both

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.

## Device 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.

## Disable 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 swapped, an `incremental establish` operation is initiated and the devices become immediately available on the link.

## Refresh 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.

### Syntax

You can issue the `swap` command for device groups, composite 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]
        [-sid SID]
```

---

### Note

`-sidSID` is required for `-sg` and `-f` operations.

---

### Options

#### **-bcv**

Targets just the BCV devices associated with the SRDF device group for the `swap` action.

#### **-all**

Target both BCV and standard devices

#### **-hop2**

Targets the SRDF action at the group's second-hop devices in a cascaded SRDF relationship.

Use alone (no options) 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
```

## Dynamic swap restrictions

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-swap 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.

## Swap 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

Use the `symrdf failover -establish` command 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 to target on page 66](#) 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).  
[Dynamic swap restrictions on page 121](#) 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 an SRDF pair \(incremental\) on page 62](#) explains this operation.

### Restrictions

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.
- This command cannot be executed 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

### NOTICE

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.

---

`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 and device files. The devices in this failover can be using synchronous or asynchronous links.

### 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.

---

### Options

#### -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.

### 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.
- 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:

- [SRDF/Asynchronous operations overview](#) ..... 126
- [SRDF/Asynchronous operations](#) ..... 129
- [Delta Set Extension management](#) ..... 139
- [Display SRDF/A](#) ..... 154

## SRDF/Asynchronous operations overview

SRDF/Asynchronous (SRDF/A) is a long distance disaster restart solution with fast application response times.

SRDF/A maintains a dependent-write consistent copy between the R1 and R2 devices across any distance with no impact to the application.

### SRDF/A 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 HYPERMAX 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 snap and clone restrictions

- 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.
- Some Snap and Clone operations are not be allowed SRDF/A-capable devices operating in asynchronous mode.  
EMC Solutions Enabler TimeFinder Family CLI User Guide provides more information.

### Move operations restrictions

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.

## 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".

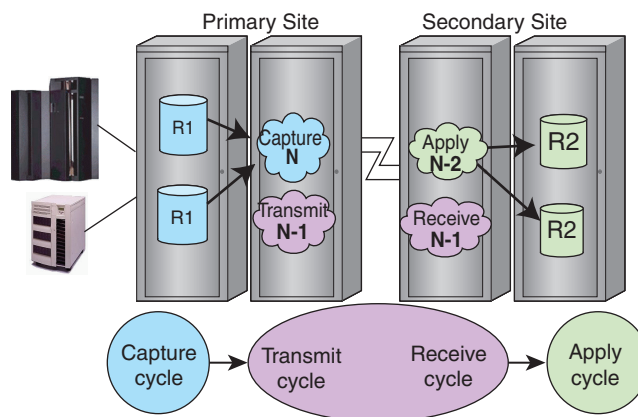
A new capture cycle cannot start until the transmit cycle completes its commit of data from the R1 side to the R2 side, and the R2 apply cycle is empty.

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 12** 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.

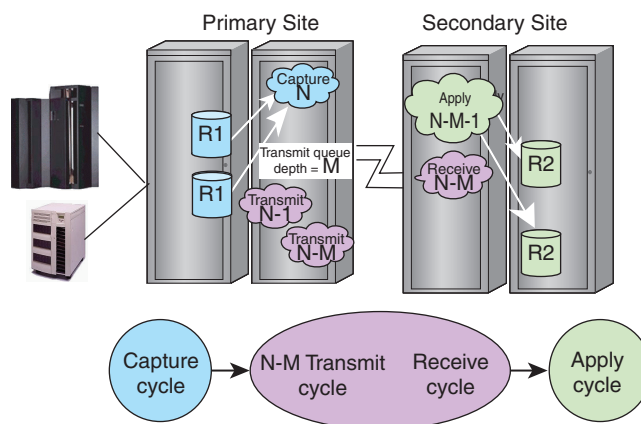
### HYPERMAX OS

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 13** 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.



## Protect the R2 side with TimeFinder BCVs

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.

## Drop SRDF/A session immediately

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.

### -immediate option 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 be 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.

The following table summarizes the operations described in this chapter.

**Table 15** SRDF/A control operations

| Control operation                                        | Command                            | Description                                                                                |
|----------------------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------|
| <a href="#">Transition replication modes on page 130</a> | <code>symrdf set mode async</code> | Change the mode of the an SRDF group, composite group or device list to asynchronous mode. |

**Table 15** SRDF/A control operations (continued)

| Control operation                                                                    | Command                                                                                                                                                             | Description                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Set SRDF/A group cycle time, priority, and transmit idle on page 132</a> | <code>symrdf set rdfa</code>                                                                                                                                        | Set the cycle time, session priority, and transmit idle for an SRDF/A group.                                                                                                                  |
| <a href="#">Check for R1 invalid tracks on page 133</a>                              | <code>symrdf verify -noinvalids -consistent</code>                                                                                                                  | Verify whether invalid tracks exist on both the R1 and R2 devices for a SRDF group, composite group or devices in a device list.                                                              |
| <a href="#">Consistency for SRDF/A devices on page 134</a>                           | <code>symrdf enable</code>                                                                                                                                          | Enable/disable consistency for a device group or devices in a device list.                                                                                                                    |
| <a href="#">Add/remove devices with the consistency exempt option on page 135</a>    | <code>symrdf createpair</code><br><code>symrdf suspend</code><br><code>symrdf movepair</code><br><code>symrdf resume</code><br><code>symrdf verify</code>           | Dynamically add and remove device pairs from an active SRDF/A session.                                                                                                                        |
| <a href="#">Display checkpoint complete status on page 138</a>                       | <code>symrdf checkpoint</code>                                                                                                                                      | Display a checkpoint complete status when the data in the current cycle is committed to the R2 side.                                                                                          |
| <a href="#">Delta Set Extension management on page 139</a>                           | <code>symrdf set rdfa_dse</code><br><code>symconfigure commit</code><br><code>symcfg show</code>                                                                    | Set the SRDF/A DSE attributes for an SRDF group.<br>Engenuity 5786 only:<br>Add/remove/enable devices in DSE pools.<br>Associate a DSE pool with an SRDF group.<br>Monitor/display DSE pools. |
| <a href="#">Activate/deactivate SRDF/A DSE on page 145</a>                           | <code>rdfa_dse_autostart</code><br><code>symrdf activate/deactivate</code>                                                                                          | Activate/deactivate SRDF/A DSE.                                                                                                                                                               |
| <a href="#">Manage transmit idle on page 147</a>                                     | <code>symrdf set rdfa -transmit_idle</code>                                                                                                                         | Allow SRDF/A sessions to manage transient link outages without dropping.                                                                                                                      |
| <a href="#">Manage SRDF/A write pacing on page 147</a>                               | <code>symrdf set rdfa_pace</code><br><code>symrdf -rdfa_pace activate</code><br><code>symrdf -rdfa_pace deactivate</code><br><code>symrdf -rdfa_wpace_exempt</code> | Enable SRDF/A write pacing for groups or devices.                                                                                                                                             |
| <a href="#">Display SRDF/A on page 154</a>                                           | <code>symdmg show</code><br><code>symrdf -g <i>DgNamequery</i> -rdfa</code>                                                                                         | Display SRDF/A sessions.<br>Display SRDF/A groups.                                                                                                                                            |
| <a href="#">List SRDF/A-capable devices on page 154</a>                              | <code>symrdf list -rdfa</code>                                                                                                                                      | List SRDF/A capable devices.                                                                                                                                                                  |

## Transition replication modes

To transition a device or group to asynchronous mode:

- Create a new device group and specify the mode as asynchronous, or
- 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.

## Transition to asynchronous mode

### Syntax

Use the `set mode async` command to 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. [SRDF set operations for TimeFinder/Snap sessions on page 450](#) and [SRDF set operations for TimeFinder/Clone sessions on page 463](#) provide more information.

---

## Transition 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 by Enginuity 5671 and higher.

**Syntax**

Use the `-consistent set mode sync` command to 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
```

## Set 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 Product 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.

**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]
```

**Options****-cycle\_time (-cyc)**

Sets 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.

**-priority (-pri)**

Sets 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.
```

## Check 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.

The `symrdf verify` command with `-noinvalids` and `-consistent` options performs an additional check to verify whether invalid tracks exist on both the R1 and R2 devices.

**Syntax**

Use the `symrdf verify` command with `-noinvalids` and `-consistent` options to verify invalid tracks on 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.
```

## 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.

## Enable consistency for SRDF/A devices

You can enable consistency for SRDF/A device pairs in a device group, storage group, or devices in a device file.

### NOTICE

For concurrent SRDF configurations, you must enable consistency for each R2 mirror separately.

### Syntax

```
symrdf -g DgName -sid SID-rdfg RdfgNumenable
symrdf -sg SgName -sid SID -rdfg RdfgNum enable
symrdf -file Filename -sid SID-rdfg RdfgNumenable
```

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 includes concurrent devices, only the R2 side specified by the `-sid SID-rdfg` argument 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 enable consistency for devices in device file `FileOne`:

```
symrdf -f FileOne -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
```

## Disable consistency for SRDF devices

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.

### Syntax

```
symrdf -g DgName -sid SID-rdfg RdfgNumdisable
symrdf -file Filename -sid SID-rdfg RdfgNumdisable
```

### Examples

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
```

## Add/remove devices with the consistency exempt option

### Note

The consistency exempt option (`-cons_exempt`) is available with Enginuity 5773.150 and higher.

Use the consistency exempt option to dynamically add and remove device pairs from an active SRDF/A session without affecting:

- The state of the session, or
- 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.

The `-cons_exempt` option can be used 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

The following procedure uses device file "Myfile" to add device pairs to an active SRDF/A session:



**Procedure**

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 the 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
```

**NOTICE**

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

The following example uses device file "Myfile" to remove device pairs from an active SRDF/A session:

**Procedure**

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
```

#### NOTICE

Do not enable host access to the R1 side until the pair state for the devices reaches Consistent.

---

## Display 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.

### Syntax

You can issue the `checkpoint` operation on a device group, composite group, storage group, or 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 [-offline]  
[-i Interval] [-c Count] checkpoint
```

### Options

#### **-c Count**

Number of times (Count) to repeat the operation before exiting.

#### **-i Interval**

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 Intervals** specified, the operation repeats continuously at the specified interval.

If **-c Count** is specified and **-i Intervals** not specified, the operation repeats the specified number of iterations using the default interval.

### Restrictions

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
```

## Delta Set Extension management

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).

### 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.

*EMC Solutions Enabler Array Management CLI User Guide* describes the steps to modify which SRP is associated with DSE.

## DSE SRP capacity management (HYPERMAX OS)

This section describes the steps to modify the capacity of the DSE SRP for arrays running HYPERMAX OS.

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.

### 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

## Modify the DSE SRP capacity

Use the `symconfigure set symmetrix dse_max_cap` command to modify the capacity of the DSE SRP.

### Syntax

```
symconfigure -sid SID commit -cmd "set symmetrix dse_max_cap =
MaxCap;"
```

### Options

#### 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.

```

## DSE pool management (Enginuity 5773 - 5876)

This section describes DSE pool management on arrays running Enginuity 5773 - 5876.

### NOTICE

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 [DSE SRP capacity management \(HYPERMAX OS\) on page 139](#)

### 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.

## Set 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.

---

For arrays running Enginuity 5876, the `symconfigure` command can also be used to set these SRDF/A DSE attributes. See the *EMC Solutions Enabler Array Management CLI User Guide*.

**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]
```

**Options****-autostart (-aut)**

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)**

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 (-ckd3390) *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.

## Clear existing DSE pool names

**Syntax**

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.

## Add 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

**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 | DISAB >];
```

**Example**

```
add dev 018B:018C to pool finance,  
type = rdfa_dse,  
member_state=ENABLE;
```

## Remove devices from an SRDF/A DSE pool

Remove SAVE devices from an SRDF/A DSE pool only if the devices are disabled and drained.

When a device is removed from a pool, it becomes available for use by other SAVE device pools.

## Syntax

```
remove dev SymDevName[:SymDevName]
from pool PoolName,
type = <snap | rdfa_dse>;
```

### Restrictions

The last device cannot be removed from an SRDF/A DSE pool if the pool is associated with an SRDF group.

### Example

```
remove dev 018B from pool finance, type = rdfa_dse;
```

## Enable/disable 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.

All the devices in an SRDF/A DSE pool cannot be disabled if the pool is currently associated with an SRDF group and SRDF/A DSE is active for the group.

## 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 using command files that are executed using the **symconfigure** command.

To set the SRDF/A DSE threshold, associate an SRDF group with a pool, and activate DSE:

### Procedure

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.

The following commands:

- Set the threshold,
- Associate with DSE pool **r1pool**,
- 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=rlpool, 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.

## Display/monitor SRDF/A DSE pool usage

Use the `symcfg show` command to display the pool utilization for a specified SRDF/A DSE pool.

### 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
```

## Activate/deactivate 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.

#### Restrictions

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.

---

## Manage 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.

### 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.

### 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
```

## Manage 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
- device-level pacing

**Group-level 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 introduced 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.

**Device-level pacing**

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.

Either or both write pacing options can be enabled or disabled. 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.

Group-level and device-level write pacing can be activated and controlled individually or simultaneously at the group, device group, composite group, or file level on the R1 side.

Both methods have 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 149](#).

**SRDF/A write pacing 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

Write-pacing behavior varies by the type of pacing, the SRDF topology (2-site, cascaded, concurrent), and OS version.

### Group-level pacing considerations

- 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 pacing considerations

- 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.

### 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.
- Sparring 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.

## 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.

## Identifying devices that cannot be paced

To identify devices that cannot be paced:

### Procedure

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    Name        CSRM TDA  time  ---  ---  Idle Time  (usecs) (%) SAU SAU  P
-----
153 (98) 1c153142  .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
.
.
.
```

## Set SRDF/A group-level write pacing attributes

To set these group attributes, the remote side must be reachable.

**Syntax**

Use the `symrdf set rdfa_pace` command to set the SRDF/A write pacing attributes for an SRDF group.

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

.....

set rdfa_pace
  [-dp_autostart {on | off}]
  [-wp_autostart {on | off}]
  [-delay 1 - 1000000]
  [-threshold 1 - 99]>
  [-both_sides]
```

**Options****-dp\_autostart (-dp\_aut)**

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)**

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.

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.

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
```

To display two entries for each attribute being applied; one for the source side and one for the target side, use the `-both_sides` option:

```
symrdf -sid 432 -rdfg 75 set rdfa_pace -delay 500 -threshold 10 -wp_autostart on -  
dp_autostart on -both_sides
```

## Activate write pacing

### 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
```

## 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.

### Restrictions

- 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
```

## Display 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 (Engenuity 5876).

---

## Show SRDF/A group information

### Syntax

Use the `symdg show DgName` command to display SRDF/A session status information.

Use the `symrdf -gDgName query -rdfa` command to display SRDF/A group information.

### Description

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

### Syntax

Use the `symrdf list -rdfa` command to list SRDF/A-capable devices (R1, R2 and R21 devices) that are currently configured in SRDF groups.

## Description

---

### 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

## SRDF/Metro Operations

This chapter describes the following topics:

- [SRDF/Metro Overview](#).....158
- [Witness SRDF groups](#)..... 161
- [Create device pairs in SRDF/Metro configurations](#)..... 162
- [SRDF/Metro CLI operations supported and restrictions](#).....166
- [Setting up SRDF/Metro \(Witness array option\)](#)..... 168
- [Display SRDF/Metro](#)..... 176

## SRDF/Metro Overview

HYPERMAX OS SR Q32015/Solutions Enabler 8.1/Unisphere 8.1 introduces SRDF/Metro.

### NOTICE

The *EMC VMAX3 Product Guide* includes more information about SRDF/Metro, including:

- SRDF/Metro life cycle
- Options for determining bias
- Failure scenarios for configurations that use Witness arrays

In traditional SRDF, R1 devices are Read/Write accessible. R2 devices are Read Only/Write Disabled.

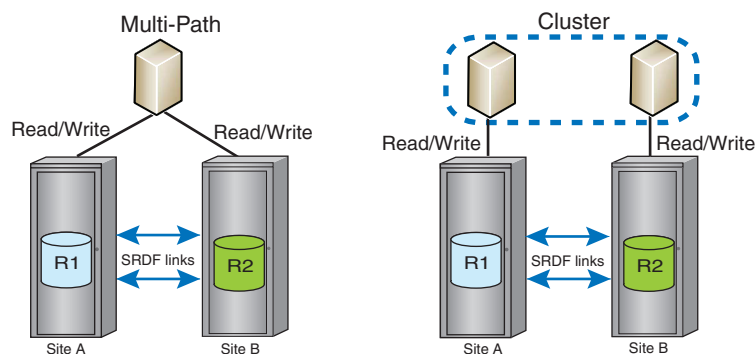
In SRDF/Metro configurations:

- R2 devices on VMAX3 arrays are Read/Write accessible to hosts.
- Hosts can write to both the R1 and R2 side of the device pair.
- R2 devices assume the same external device identity (geometry, device WWN) as their R1.

This shared identity causes the R1 and R2 devices to appear to hosts(s) as a single virtual device across the two arrays.

SRDF/Metro can be deployed with either a single multi-pathed host or with a clustered host environment.

**Figure 14** SRDF/Metro



Hosts can read and write to both the R1 and R2 devices.

For single host configurations, host I/Os are issued by a single host. Multi-pathing software (for example, PowerPath) directs parallel reads and writes to both arrays.

For clustered host configurations, host I/Os can be issued by multiple hosts accessing both sides of the SRDF device pair. Each cluster node has dedicated access to an individual VMAX3 array.

In both single host and clustered configurations, writes to the R1 or R2 devices are synchronously copied to the paired device. Write conflicts are resolved by the SRDF/Metro software to maintain consistent images on the SRDF devices pairs. The R1 device and its paired R2 device appear to the host as a single virtualized device.

SRDF/Metro is managed using Solutions Enabler 8.1 and greater.

SRDF/Metro requires a license on both VMAX3 arrays.

VMAX3 arrays can simultaneously support SRDF groups configured for SRDF/Metro operations and SRDF groups configured for traditional SRDF operations.

### Key differences in SRDF/Metro

- In SRDF/Metro configurations:
  - R2 device is Read/Write accessible to the host.
  - Host(s) can write to both R1 and R2 devices.
  - Both sides of the SRDF device pair appear to the host(s) as the same device.
  - The R2 device assumes the personality of the primary R1 device (geometry, device WWN, etc.).
  - Introduces two new RDF pair states:
    - ActiveActive for configurations using the Witness array option
    - ActiveBias for configurations using bias

---

#### Note

R1 and R2 devices should not be presented to the cluster until they reach one of these 2 states and present the same WWN.

---

- For SRDF/Metro configurations that are part of a CG, CG control and set operations are allowed on only one group at a time.
- All device pairs in an SRDF/Metro group are managed together for all supported operations, with the following exceptions:
  - If all the SRDF device pairs are Not Ready (NR) on the link, createpair operations can add devices to the group if the new device pairs are created Not Ready (NR) on the link.
  - If all the SRDF device pairs are Not Ready (NR) on the link, deletepair operations can delete a subset of the SRDF devices in the SRDF group.
- Devices have an additional attribute; bias, described below.

### Bias

SRDF/Metro introduces the concept of *bias*. Bias determines which side of the SRDF device pair remains R/W accessible to the host in the event of a link or other failure. The *bias device* refers to the device that will remain accessible to the host when the RDF link is Not Ready (NR). The bias device will be exported as the R1 in all SE displays and commands. The non-bias device will be exported as the R2 in all SE displays and commands.

In the event of link or other failures, SRDF/Metro uses one of two options to determine which side of an SRDF device pair remains R/W accessible to the host. The device side that remains R/W accessible to the host is called the *bias* side. The two options to determine bias are:

- **Bias option:** Device pairs for SRDF/Metro are created with a new option, `-use_bias`, to indicate that the bias method should be used to determine which side of the device pair remains R/W accessible if the link fails or if the devices are made NR on the link (suspended). In either of those situations, the "bias" side remains R/W accessible to the hosts, and the non-bias side becomes inaccessible to the hosts. The `createpair` operation sets the R1 to be the bias side. The pair state for device pairs in an SRDF/Metro using the bias option is ActiveBias.
- **Witness array option:** HYPERMAX OS or Enginuity on a third (Witness) array monitors SRDF/Metro, determines the type of failure, and uses the information to choose one side of the device pair to remain R/W accessible to the host. The state of the SRDF/

Metro configuration at the time the SRDF links go Not Ready determines which side will be the bias side.

The pair state for device pairs in an SRDF/Metro using the Witness array option is ActiveActive.

The Witness array option is the default.

- When all RDF device pairs in the SRDF group have reached the ActiveActive or ActiveBias pair state, the user can change the bias side.

## Witness array option

By default SRDF/Metro uses a third "Witness" array to determine the bias side when the link is suspended or there is a link failure. A Witness array is not the R1 or the R2 array, and must be running one of the following:

- Engenuity 5876 with ePack containing fixes to support N-x connectivity
- HYPERMAX OS 5977 Q12015SR with ePack containing fixes to support N-x connectivity
- HYPERMAX OS 5977 Q32015SR or above

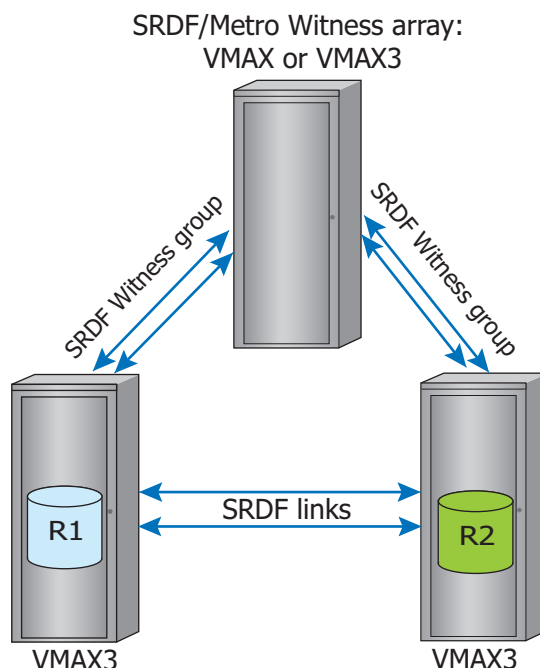
The Witness array must have SRDF connectivity to both the R1 side array and R2 side array.

SRDF remote adapters (RA's) are required on the Witness array with applicable network connectivity to both the R1 side and R2 side arrays. When the Witness array is identified to both the SRDF/Metro paired arrays, the configuration enters Witness Protected state.

For complete redundancy, multiple Witness arrays may be configured. If the auto configuration process fails and no other applicable Witness arrays are identified, SRDF/Metro falls back to the bias option.

The Witness option requires 2 SRDF groups; one between the R1 array and the Witness array, and a second between the R2 array and the Witness array:

**Figure 15** SRDF/Metro Witness array and groups



Solutions Enabler ensures that the Witness groups exist and are online when establish (including create -establish/--restore) or restore operations are started. SRDF/Metro auto-



detects which Witness an SRDF/Metro group is using, so there is no need to specify the Witness.

If no Witness is available, the operation fails.

When the SRDF/Metro device pairs are configured to use a Witness array, and the Witness array is accessible, the state of the device pairs is ActiveActive.

If the Witness array becomes inaccessible from both the R1 and R2 arrays, HYPERMAX OS sets the R1 side as the bias side, the R2 side as the non-bias side, and the state of the device pairs is ActiveBias.

## Witness SRDF groups

The Witness array option requires two Witness SRDF groups:

- One between the R1 array and the Witness array
- One between the R2 array and the Witness array

Only one Witness SRDF group can be created between any two arrays.

Witness SRDF groups must be empty. Creation of SRDF device pairs is blocked in Witness SRDF groups.

At least one side of the Witness SRDF group must be running HYPERMAX OS Q3 2015SR or above.

### Witness SRDF group attributes

Some attributes of Witness SRDF groups are different than a standard SRDF group. Differences include:

- Link limbo - The default value for an Witness SRDF group is 1 second. EMC recommends that this value not be increased, as doing so decreases Witness protection.

## Add an SRDF/Metro Witness group

To create a SRDF/Metro Witness group, specify the `-witness` option for the `addgrp` operation.

To create a Witness group Witness1 between group 10 on array 0085 and group 110 on array 086:

```
symrdf addgrp -sid 0085 -rdfg 10 -remote_sid 086 -remote_rdfg 110 -dir 1g:28
  -remote_dir 1g:28 -nop -label Witness1 -witness
```

## Remove an SRDF/Metro Witness group

A SRDF/MetroWitness group cannot be removed when at least one SRDF/Metro SRDF group is protected by that Witness group.

To remove a Witness group, specify the `-witness` option for the `removegrp` operation.

To remove SRDF/Metro Witness group 10:

```
symrdf removegrp -sid 0085 -rdfg 10 -nop -witness
```

## Modify an SRDF/Metro Witness group

To modify a SRDF/Metro Witness group, specify the `-witness` option for the `modifygrp` operation.

To add director 1g:29 to SRDF/Metro Witness group 10:

```
symrdf modifygrp -add -sid 0085 -rdfg 10 -dir 1g:29 -witness
```

## Suspend an SRDF/Metro group

The `suspend` action suspends I/O traffic on the SRDF links for the specified remotely mirrored SRDF pairs in the group or device file and makes them Not Ready (NR) on the SRDF link.

In SRDF/Metro configurations, where bias determines which side of the device pair remains accessible to the host, you can use the `-bias R1|R2` option to change the bias side in conjunction with the `suspend` operation.

---

### Note

`-bias -R1` does not change the bias. Bias remains set on the current R1 side.

---

The following restrictions apply to `suspend` in SRDF/Metro configurations:

- The `suspend` operation must include all devices in the group.
- If the Witness method determines which side of the device pair remains accessible to the host, the Witness groups must be online.
- `-force` is required to suspend a SRDF/Metro group.

To suspend the SRDF links for devices in the specified device file in group 86 and set bias to the R2 side:

```
symrdf -f /tmp/device_file -sid 085 -rdfg 86 suspend -force -bias R2
```

## Create device pairs in SRDF/Metro configurations

To create SRDF devices in a new or existing SRDF/Metro configuration, the `-rdf_metro` option must be used with the `createpair` command.

SRDF/Metro configurations do not support concurrent or cascaded SRDF configurations. Thus, the devices specified for the `createpair` operation must be non-RDF devices.

If the SRDF group in which the device pairs are being created is not empty, all device pairs already in the SRDF group must be Not Ready (NR) on the SRDF link.

Even if the device pairs are being created in an existing SRDF/Metro group, the `-rdf_metro` option is required.

Use the `-use_bias` option to indicate that the SRDF/Metro configuration will use bias rather than Witness protection. The R1 side will be the bias side.

---

### Note

`-use_bias` can only be used with the `-establish` or `-restore` options.

---

For configurations that use the Witness array option, 2 Witness SRDF groups must be defined before the `createpair` operation:

- One Witness SRDF group between the R1 side array and the Witness array
  - One Witness SRDF group between the R2 side array and the Witness array
- The two Witness groups must be online or the `createpair` operation fails.

### Examples

In the following example:

- `-rdf_metro` indicates the devices are created in a SRDF/Metro configuration.
- `-sid 174 -type R1` indicates array 174 is the R1 side.
- `-sg` specifies the name of the storage group.
- `-remote_sg` specifies the remote storage group name.
- `-establish` starts the synchronization process from R1 to R2 devices.

```
symrdf createpair -rdf_metro -sid 174 -type R1 -rdfig 2 -sg RDF1_SG
               -remote_sg RDF2_SG -establish
```

```
Execute an RDF 'Create Pair' operation for storage group 'RDF1_SG' (y/[n]) ? y
```

```
An RDF 'Create Pair' operation execution is
in progress for storage group 'RDF1_SG'. Please wait...
```

```
Create RDF Pair in (0174,002).....Started.
Create RDF Pair in (0174,002).....Done.
Mark target device(s) in (0174,002) for full copy from source....Started.
Devices: 006B-0074 in (0174,002).....Marked.
Mark target device(s) in (0174,002) for full copy from source....Done.
```

In the following example, the `createpair` command:

- Creates device pairs using device pairs listed in a device file `/tmp/device_file`,
- Specifies the pairs are in a SRDF/Metro configuration (`-rdf_metro`)

```
symrdf -f /tmp/device_file -sid 085 -type R1 -rdfig 86 createpair -
est -rdf_metro
```

## Create SRDF/Metro pairs with the `-establish` option

Before you begin:

- All devices in the group must be specified for the operation. That is, the group must be empty prior to the `createpair -rdf_metro -establish` operation
- The `-rdf_metro` option must be specified to create devices.
- If the bias method of determining which side of the device pair remains accessible to the host is used, the `-use_bias` option must be specified.
- If the Witness method of determining which side of the device pair remains accessible to the host is used, the Witness groups must be online.
- The operation creates the device pairs and makes them RW on the link. When the `createpair` operation completes, the device pair's mode is Active and pair state is `SyncInProg`.
- The pair state is `SyncInProg` until there are no invalids and the R2 side has acquired the R1 device information. Then the pair state transitions to `ActiveActive` or `ActiveBias`.

**Example - Create SRDF/Metro pairs (Witness)**

To create SRDF/Metro device pairs using device file `device_file`:

```
symrdf -f /tmp/device_file -sid 085 -type r1 -rdfg 86 createpair
        -establish -rdf_metro
```

**Example - Create SRDF/Metro pairs (bias)**

To create SRDF/Metro device pairs using device file `device_file` and specify the bias method:

```
symrdf -f /tmp/device_file -sid 085 -type r1 -rdfg 86 createpair
        -establish -rdf_metro -use_bias
```

**Create SRDF/Metro pairs with the -invalidate option****Syntax**

Use the `symrdf createpair` command with the `-invalidate r1` or `-invalidate r2` option to create devices (R1 or R2) in a new or existing configuration.

The `-rdf_metro` option must be specified to create the specified (R1 or R2) target device.

The `createpair -rdf_metro -invalidate R1/R2` operation can be used to add device pairs to an existing SRDF/Metro configuration, provided that all device pairs already in the group are Not Ready (NR) on the SRDF link.

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.
- Use the `restore` command to start copying to the invalidated source device.

**Example**

```
symrdf createpair -sid 55 -file devicefile -rdfg 1 -type R1 -invalidate r2 -rdf_metro
```

**Create SRDF/Metro pairs with the -restore option**

Use the `-restore` option to copy data back to the R1 source devices.

- All devices in the group must be specified for the operation. The group must be empty prior to the `createpair -rdf_metro -restore` operation.
- The `-rdf_metro` option must be specified to create devices in a new or existing configuration.
- If the bias option determines which side remains accessible to the host in the event of a link or other failure, the `-use_bias` option must be specified.
- If the Witness option determines which side remains accessible to the host in the event of a link or other failure, the Witness groups must be online.
- The operation creates the device pairs and makes them RW on the link. When the `createpair` operation completes, the device pair's mode is Active and their pair state is `SyncInProg`.
- The pair state is `SyncInProg` until there are no invalids and the R2 side has acquired the R1 device information. Then the pair state transitions to `ActiveActive` or `ActiveBias`.

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.

### Restrictions

- Only non-SRDF devices can be specified for a createpair into an SRDF/Metro configuration
- If SRDF Witness groups are used, they must be present and online.
- SRDF device pairs cannot be created in an SRDF Witness group
- Both the R1-side and R2-side arrays must be running HYPERMAXOS 5977 Q32015SR or later.
- The devices cannot have GCM set.
- The R2 cannot be larger than the R1.
- The devices cannot have User Not Ready set.
- The devices cannot have User Geometry set.
- The devices cannot be BCVs.
- An SRDF/Metro group cannot contain a mixture of R1 and R2 devices.

### Example - Create SRDF/Metro pairs (Witness)

To create SRDF/Metro device pairs using device file `device_file`:

```
symrdf -f /tmp/device_file -sid 085 -type r1 -rdfg 86 createpair
      -restore -rdf_metro
```

### Example - Create SRDF/Metro pairs (bias)

To create SRDF/Metro device pairs using device file `device_file` and specify the bias method:

```
symrdf -f /tmp/device_file -sid 085 -rdfg 86 createpair -restore -rdf_metro -use_bias
```

## Delete SRDF/Metro pairs

### Delete both sides of an SRDF/Metro pair

The `deletepair` operation:

- Deletes the SRDF/Metro device pairing,
- Removes the pairing information from the array and the SYMAPI database
- Changes the specified devices to non-SRDF devices
- If the operation applies to a group, changes the group to a regular device group

[Deleting dynamic SRDF device pairs on page 113](#) provides more information about deleting pairs.

### NOTICE

Deleting the last device pair from an SRDF group in an SRDF/Metro configuration terminates the SRDF/Metro configuration, after which the group can be re-used either for another SRDF/Metro configuration or for a traditional SRDF configuration.

### Delete one side of an SRDF/Metro pair

The `half_deletepair` operation removes 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 (`-f FileName`), device group (`-g GrpName`), consistency group (`-cg CGrpName`), or storage group (`-sg SGrpName`).

#### NOTICE

If a `half_deletepair` operation removes all devices from one side of an SRDF group that is in an SRDF/Metro configuration, that side of the group will no longer be part of the SRDF/Metro configuration.

---

## SRDF/Metro CLI operations supported and restrictions

This section describes:

- Changes to the CLI to support SRDF/Metro configurations
- Commands to restore device personality
- Operations that are not allowed in SRDF/Metro configurations
- Operations blocked when device or group is in a SRDF/Metro configuration
- Other restrictions applicable to SRDF/Metro configurations

### addgrp/removegrp/modifygrp commands

For SRDF/Metro configurations, a new option, `-witness`, has been added to the `addgrp`, `removegrp`, and `modifygrp` commands to enable the creation of Witness SRDF groups. The `-witness` option is required to create, modify or remove a Witness SRDF group.

### createpair command

For SRDF/Metro configurations, new options, `-rdf_metro` and `-use_bias` have been added to support the creation of a SRDF/Metro device pairs. The `createpair -rdf_metro` command supports the following operations:

- `-establish [-use_bias]`
- `-restore [-use_bias]`
- `-invalidate r1`
- `-invalidate r2`

### Commands to restore device personality

When a device is removed from an SRDF/Metro configuration, the device retains its federated personality. The device's original native personality can be restored. A new option set `-no_identity` has been added to the following commands to support restoring devices in a DG or CG:

- `symdev`
- `symmsg`
- `symdg`
- `symcg`

### SRDF operations not supported in SRDF/Metro configurations

The following operations are not supported for SRDF devices in an SRDF/Metro configuration:

- `activate <-rdfa_dse | -rdfa_pace | -rdfa_wpace | -rdfa_devpace | -rdfa_wpace_exempt>`
- `checkpoint`

- deactivate <-rdfa\_dse |-rdfa\_pace |-rdfa\_wpace |-rdfa\_devpace  
|-rdfa\_wpace\_exempt>
- disable
- enable
- failback
- failover
- invalidate <R1 | R2>
- merge
- migrate
- movepair/half\_movepair
- msc\_cleanup
- not\_ready <R1 | R2>
- ready <R1 | R2>
- refresh <R1 | R2>
- resume
- rw\_enable <R1 | R2>
- rw\_disable R2
- set mode
- split
- swap
- update
- write\_disable <R1 | R2>

#### **Operations blocked when device or group is in an SRDF/Metro configuration**

The following commands are blocked if the SRDF device or group is part of an

SRDF/Metro configuration:

- symconfigure create dev/add rdf mirror/remove rdf mirror
- symconfigure convert dev  
Converting the following device configurations is blocked if the SRDF device is part of an  
SRDF/Metro configuration:
  - TDEV
  - BCV+TDEV
  - RDF1-BCV+TDEV
  - RDF2-BCV+TDEV
- symcg enable/remove
- symcg modify if the devices in the staging area are part of a SRDF/  
Metroconfiguration.
- symrcopy - Devices in SRDF/Metro cannot be used as ORS control devices.
- symreplicate start/setup/restart
- symstar setup
- symstar modifycg if the devices in the staging area are part of a SRDF/  
Metroconfiguration.

**Additional SRDF/Metro restrictions**

The following restrictions apply to devices in SRDF/Metro configurations:

- SRDF/Metro configurations do not allow the R2 device to be larger than the R1.
- Only non-SRDF devices can become part of an SRDF/Metro configuration.
- The -format, -remote, and -rdf\_mode options of the createpair operation are not supported.
- Devices that are part of an SRDF/Metro configuration cannot:
  - Have User Not Ready set
  - Have User Geometry set
  - Be monitored by SRDF Automated Recovery
  - Be migrated
  - Be added to SRDF CGs
  - Be part of a concurrent configuration
  - Be part of a cascaded configuration
  - Be part of an SRDF/Star configuration

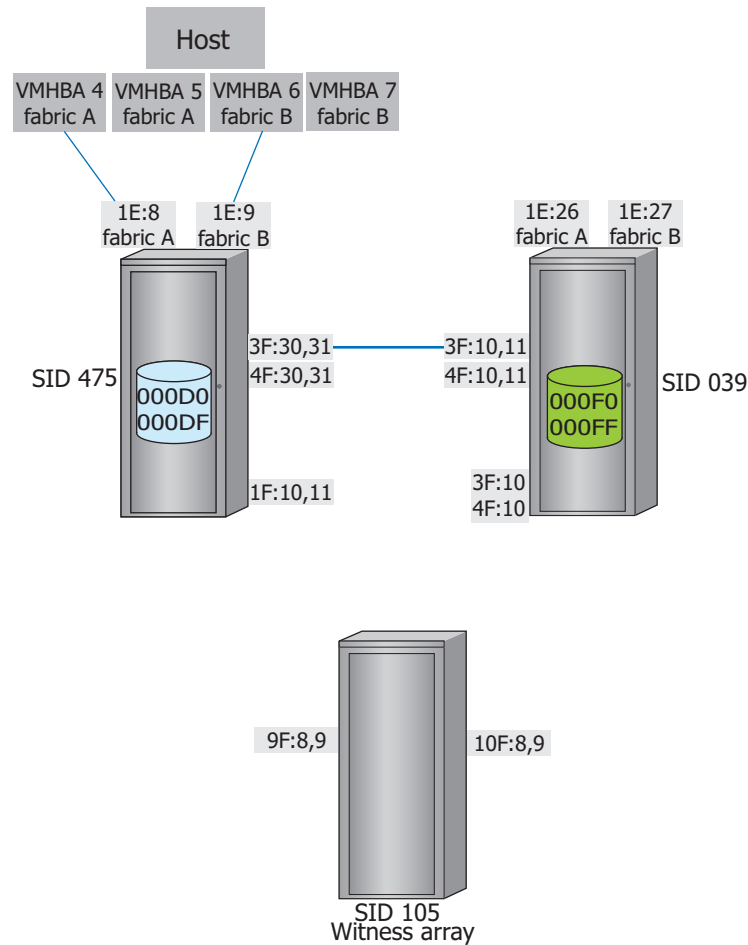
## Setting up SRDF/Metro (Witness array option)

This sample procedure shows the steps to set up SRDF/Metro using a Witness array to determine which side of the SRDF device pair remains accessible to the host in the event of a link or other failure. In the following image:

- The array that will become the R1 side is mapped/masked to the host.
- The array that will become the R2 side is NOT mapped/masked to the host.

**Figure 16** Setting up SRDF/Metro with Witness array; Before





### Procedure

1. On the host, use the `symcli` command to verify the version of Solutions Enabler is 8.1.
2. If a single host is being used for the presentation of both R1 and R2 devices, use the `verify` command to confirm that multi-pathing software such as PowerPath or similar native functionality is in use.
3. Use the `symrdf addgrp` command to create Witness SRDF groups between SIDs 475/105 and 039/105:

```
symrdf addgrp -witness -label SG_120 -sid 000196700475 -rdfg 120 -dir 1F:10,1F:11 -
remote_sid 000197200105 -remote_rdfg 120 -remote_dir 9F:8,9F:9
```

Successfully Added Dynamic RDF Group 'SG\_120' for Symm: 000196700475

```
symrdf addgrp -witness -label SG_121 -sid 000197200039 -rdfg 121 -dir 3F:10,4F:10 -
remote_sid 000197200105 -remote_rdfg 121 -remote_dir 10F:8,10F:9
```

Successfully Added Dynamic RDF Group 'SG\_121' for Symm: 000197200039

4. Use the `symrdf addgrp` command to create SRDF group for the SRDF pairs between SIDs 475 and 039:

```
symrdf addgrp -label SG_20 -sid 000196700475 -rdfg 20 -dir 3F:30,3F:31,4F:30,4F:31 -
remote_sid 000197200039 -remote_rdfg 20 -remote_dir 3F:10,3F:11,4F:10,4F:11
```

Successfully Added Dynamic RDF Group 'SG\_20' for Symm: 000196700475

5. Use `symcfg list` commands with the `-rdf_metro` option to display the SRDF groups.

To display group 20 on SID 475:

**symcfg - sid 475 -rdfg 20 -rdf\_metro list**

Symmetrix ID : 000196700475

| S Y M M E T R I X |     |         |              | R D F |       | G R O U P S |      |     | R D F M e t r o |   |              |
|-------------------|-----|---------|--------------|-------|-------|-------------|------|-----|-----------------|---|--------------|
| Local             |     | Remote  |              |       |       | Group       |      |     |                 |   |              |
| RA-Grp            | sec | RA-Grp  | SymmID       | ST    | Name  | LPDS        | CHTM | Dir | CE              | S | Witness      |
| 20 (13)           | 10  | 20 (13) | 000197200039 | OD    | SG_20 | XX..        | ..XX | F-S | WW              | N | 000197200105 |

Legend:

Group (S)tatus : O = Online, F = Offline  
 Group (T)ype : S = Static, D = Dynamic, W = Witness  
 Director (C)onfig : F-S = Fibre-Switched, F-H = Fibre-Hub  
 G = GIGE, E = ESCON, T = T3, - = N/A

Group Flags :  
 Prevent Auto (L)ink Recovery : X = Enabled, . = Disabled  
 Prevent RAs Online Upon (P)ower On: X = Enabled, . = Disabled  
 Link (D)omino : X = Enabled, . = Disabled  
 (S)TAR/SQAR mode : N = Normal, R = Recovery, . = OFF  
 S = SQAR Normal, Q = SQAR Recovery

RDF Software (C)ompression : X = Enabled, . = Disabled, - = N/A  
 RDF (H)ardware Compression : X = Enabled, . = Disabled, - = N/A  
 RDF Single Round (T)rip : X = Enabled, . = Disabled, - = N/A  
 RDF (M)etro : X = Configured, . = Not Configured

RDF Metro Flags :  
 (C)onfigured Type : W = Witness, B = Bias, - = N/A  
 (E)ffective Type : W = Witness, B = Bias, - = N/A  
 Witness (S)tatus : N = Normal, D = Degraded, F = Failed, - = N/A

To display group 20 on SID 039:

**symcfg -sid 039 -rdfg 20 -rdf\_metro list**

Symmetrix ID : 000197200039

| S Y M M E T R I X |     |         |              | R D F |       | G R O U P S |      |     | R D F M e t r o |   |              |
|-------------------|-----|---------|--------------|-------|-------|-------------|------|-----|-----------------|---|--------------|
| Local             |     | Remote  |              |       |       | Group       |      |     |                 |   |              |
| RA-Grp            | sec | RA-Grp  | SymmID       | ST    | Name  | LPDS        | CHTM | Dir | CE              | S | Witness      |
| 20 (13)           | 10  | 20 (13) | 000196700475 | OD    | SG_20 | XX..        | ..XX | F-S | WW              | N | 000197200105 |

To display all groups on SID 105:

**symcfg -sid 105 -rdfg all list**

Symmetrix ID : 000197200105

S Y M M E T R I X R D F G R O U P S

| Local |  | Remote |  | Group |       |     | RDFA Info |       |  |
|-------|--|--------|--|-------|-------|-----|-----------|-------|--|
| LL    |  |        |  |       | Flags | Dir | Flags     | Cycle |  |
|       |  |        |  |       |       |     |           |       |  |

| RA-Grp   | sec | RA-Grp   | SymmID       | ST | Name   | LPDS | CHTM | Cfg | CSRM | time | Pri |
|----------|-----|----------|--------------|----|--------|------|------|-----|------|------|-----|
| 120 (77) | 1   | 120 (77) | 000196700475 | OW | SG_120 | XX.. | ..X. | F-S | -IS- | 15   | 33  |
| 121 (78) | 1   | 121 (78) | 000197200039 | OW | SG_121 | XX.. | ..X. | F-S | -IS- | 15   | 33  |

6. Use the `createpair` command with the `-rdf_metro` option to create SRDF/Metro device pairs.

To create SRDF/Metro device pairs in local group 20 and remote group 20:

```
symrdf -sid 000196700475 -rdfg 20 -f rdfg20 createpair -type r1 -rdf_metro -establish
```

An RDF 'Create Pair' operation execution is in progress for device file 'rdfg20'. Please wait...

```
Create RDF Pair in (0475,020).....Started.
Create RDF Pair in (0475,020).....Done.
Mark target device(s) in (0475,020) for full copy from source....Started.
Devices: 00D0-00D7 in (0475,020).....Marked.
Mark target device(s) in (0475,020) for full copy from source....Done.
Merge track tables between source and target in (0475,020).....Started.
Devices: 00D0-00D7 in (0475,020).....Merged.
Merge track tables between source and target in (0475,020).....Done.
Resume RDF link(s) for device(s) in (0475,020).....Started.
Resume RDF link(s) for device(s) in (0475,020).....Done.
```

The RDF 'Create Pair' operation successfully executed for device

7. Wait for the device pairs to reach ActiveActive state:

```
symrdf -sid 000196700475 -rdfg 20 -f rdfg20 verify -activeactive -i 15
```

None of the device(s) in the list are in 'ActiveActive' state.

All device(s) in the list are in 'ActiveActive' state.

8. Query the device pairs:

```
symrdf -sid 000196700475 -rdfg 20 -f rdfg20 query
```

```
Symmetrix ID          : 000196700475      (Microcode Version: 5977)
Remote Symmetrix ID    : 000197200039      (Microcode Version: 5977)
RDF (RA) Group Number  : 20 (13)

Source (R1) View      Target (R2) View      MODE
-----
Standard              LI              ST
A                      N              A
Logical              K Sym          T R1 Inv  R2 Inv
Sym                  S Dev          E Tracks Tracks
Device              Dev
E Tracks            Tracks
MACE
STATE

N/A      000D0 RW          0          0 RW 000F0 RW          0          0 T.X. ActiveActive
N/A      000D1 RW          0          0 RW 000F1 RW          0          0 T.X. ActiveActive
N/A      000D2 RW          0          0 RW 000F2 RW          0          0 T.X. ActiveActive
N/A      000D3 RW          0          0 RW 000F3 RW          0          0 T.X. ActiveActive
N/A      000D4 RW          0          0 RW 000F4 RW          0          0 T.X. ActiveActive
N/A      000D5 RW          0          0 RW 000F5 RW          0          0 T.X. ActiveActive
N/A      000D6 RW          0          0 RW 000F6 RW          0          0 T.X. ActiveActive
N/A      000D7 RW          0          0 RW 000F7 RW          0          0 T.X. ActiveActive

Total
Track(s)          0          0          0          0
MB(s)            0.0          0.0          0.0          0.0
```

9. After the pairs have reached ActiveActive state, display the device WWNs to verify the R1 and R2 have the same WWN:

```
symdev list -sid 475 -wnn -devs d0:d3
```

```
Symmetrix ID: 000196700475
Device Name
```

```
Device
```

| Sym   | Physical    | Config    | Attr | WWN                              |
|-------|-------------|-----------|------|----------------------------------|
| 000D0 | Not Visible | RDF1+TDEV |      | 60000970000196700475533030304430 |
| 000D1 | Not Visible | RDF1+TDEV |      | 60000970000196700475533030304431 |
| 000D2 | Not Visible | RDF1+TDEV |      | 60000970000196700475533030304432 |
| 000D3 | Not Visible | RDF1+TDEV |      | 60000970000196700475533030304433 |

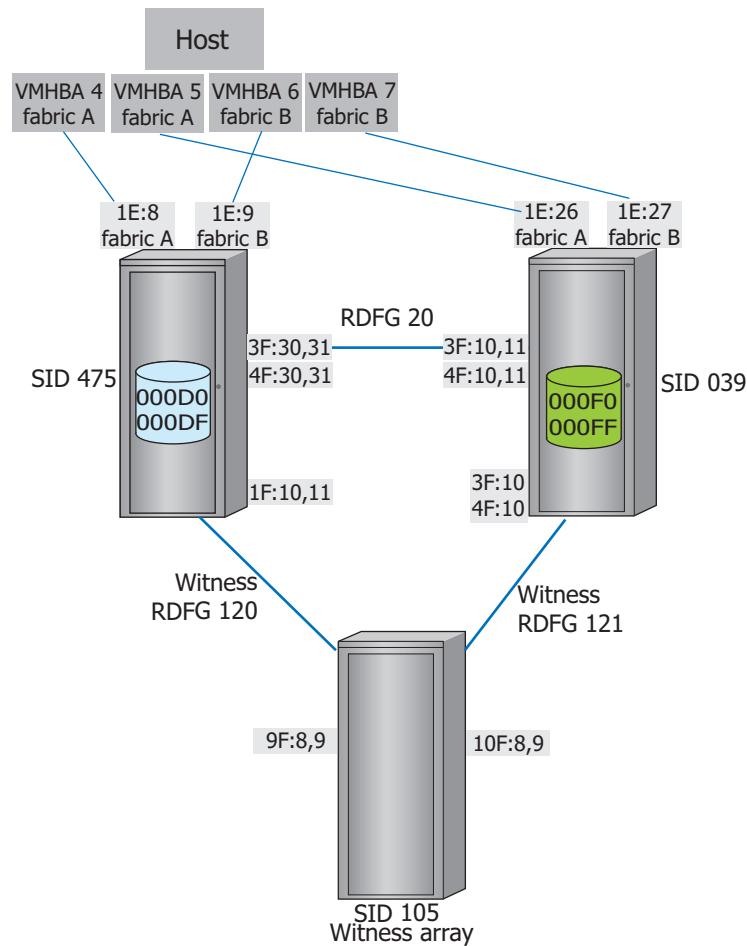
```
symdev list -sid 039 -wnn -devs f0:f3
```

```
Symmetrix ID: 000197200039
Device Name
```

```
Device
```

| Sym   | Physical    | Config    | Attr | WWN                              |
|-------|-------------|-----------|------|----------------------------------|
| 000F0 | Not Visible | RDF2+TDEV |      | 60000970000197200039533030304630 |
| 000F1 | Not Visible | RDF2+TDEV |      | 60000970000197200039533030304631 |
| 000F2 | Not Visible | RDF2+TDEV |      | 60000970000197200039533030304632 |
| 000F3 | Not Visible | RDF2+TDEV |      | 60000970000197200039533030304633 |

10. Map and mask the R2 array to the host and access additional paths to the devices.  
The following image shows the new configuration.

**Figure 17** Setting up SRDF/Metro with Witness array; After

## Setting SRDF/Metro bias

In an SRDF/Metro configuration, the SRDF link between the two sides of the SRDF device pair is used to ensure consistency of the data on the two sides.

If the SRDF device pair becomes Not Ready (NR) on the SRDF link, SRDF/Metro allows one side to remain accessible to the host(s), while it makes the other side inaccessible to the host(s).

The choice of which side remains accessible to the host is called "bias". Refer to the *VMAX3 Product Guide* for a detailed description of bias.

By default, the `createpair -rdf_metro` operation places an SRDF device pair into an SRDF/Metro configuration and pre-configures the bias to the R1 side of the pair.

You can change the bias once all SRDF device pairs in the SRDF group have reached the ActiveActive or ActiveBias SRDF pair state. The bias side is represented as R1 and the non-bias side is represented as R2. Changing the bias changes the SRDF personalities of the two sides of the SRDF device pair.

The `symrdf` command has been enhanced with a new set option `bias R1 | R2` to support changing the bias side of devices in a device file, device group, composite group, or storage group.

**Procedure**

1. Changing the bias makes it appear that a `symrdf swap` has been performed. Use the `symrdf query` command to display the target devices before changing their bias.
2. Use the `symrdf set bias` command to change the bias of devices in the target device file, device group, composite group, or storage group.

To change the bias of devices in storage group `RDF1_SG` to `R2`:

```
symrdf -sid 174 -sg RDF1_SG -rdfig 2 set bias R2
```

```
Execute an RDF Set 'Bias R2' operation for storage
group 'RDF1_SG' (y/[n]) ? y
```

```
An RDF Set 'Bias R2' operation execution is in
progress for storage group 'RDF1_SG'. Please wait...
```

```
The RDF Set 'Bias R2' operation successfully executed
for storage group 'RDF1_SG'.
```

3. Use the `symrdf query` command to confirm the change.

## Setting bias when the group is suspended

The bias may also be changed when the group is suspended.

**Procedure**

1. Use the `symrdf suspend` command with the `-bias R2` option to suspend the SRDF group while changing the bias to the `R2` side:

**NOTICE**

The `-force` option is required to complete this operation because the devices are enabled.

```
symrdf -sid 174 -sg RDF1_SG -rdfig 2 suspend -bias R2 -force
```

```
Execute an RDF 'Suspend' operation for storage
group 'rdf1_sg' (y/[n]) ? y
```

```
An RDF 'Suspend' operation execution is
in progress for storage group 'rdf1_sg'. Please wait...
```

```
Suspend RDF link(s) for device(s) in (0174,002).....Done.
```

```
The RDF 'Suspend' operation successfully executed for
storage group 'rdf1_sg'.
```

2. Use the `symrdf establish` command with the `-bias R1` option to resume the link. The bias remains set on the `R1` side:

```
symrdf -sid 174 -sg RDF1_SG -rdfig 2 establish -bias R1 -force
```

```
Execute an RDF 'Incremental Establish' operation for storage
group 'rdf1_sg' (y/[n]) ? y
```

```
An RDF 'Incremental Establish' operation execution is
in progress for storage group 'rdf1_sg'. Please wait...
```

```
Suspend RDF link(s) for device(s) in (0174,002).....Done.
Resume RDF link(s) for device(s) in (0174,002).....Started.
Read/Write Enable device(s) in (0174,002) on SA at target (R2)...Done.
```

The RDF 'Incremental Establish' operation successfully initiated for storage group 'rdf1\_sg'.

## Deactivate SRDF/Metro

When the SRDF device pairs are in the suspended SRDF pair state, use the `deletepair` operation to terminate the SRDF/Metro configuration at both sides of the SRDF group. When all devices in the SRDF/Metro group have been deleted, that group will no longer be part of an SRDF/Metro configuration.

### NOTICE

`deletepair` may also be performed on a subset of the SRDF device pairs in the SRDF group. Any remaining devices continue to be part of the existing SRDF/Metro configuration

### Planned outage

For a planned outage, use the `suspend` operation to take the device pairs out of the ActiveActive or ActiveBias SRDF pair state and put them into a suspended SRDF pair state.

When the outage is complete, do one of either:

- Use the `establish` command to keep the data on the R1 side.
- Use the `restore` command to keep the data on the R2 side.

### Unplanned outage

During an unplanned outage, when the SRDF device pairs are in a suspended or partitioned SRDF pair state, use the `half_deletepair` operation to terminate the SRDF/Metro configuration at one side of the SRDF group. `half_deletepair` may be performed on all or on a subset of the SRDF devices on one side of the SRDF group. After a `half_deletepair` operation:

- The devices affected will no longer be SRDF devices.
- The devices at the other side of the SRDF group continue to be configured for SRDF/Metro.

## Restoring native device personality

When an SRDF/Metro pair is RW on the SRDF link and has reached the ActiveActive or ActiveBias pair state, both sides of the SRDF device pair share the ID that the R1 device advertised at the time the devices were made RW on the link. This device ID is "owned" by the bias side of the device pair, originally the R1 side.

A `set bias R2` or `suspend -bias r2` operation transfers ownership of the device pair's ID to the R2 side.

After a `deletepair` operation, the device side that last owned the ID (the bias side, referred to as the R1 in displays and exported data) uses that ID. The other device side (non-bias side) will use the other ID.

If the bias side of a device pair was never changed, the R2 side goes back to its original device ID after a `deletepair` operation.

If the bias side was changed, both sides of the device pair retain the shared device ID after a `deletepair` operation.

The original native personality can be restored for devices that retain the shared ID after a `deletepair` operation. The following rules and restrictions apply to restoring the native personality of a device which has a federated personality as a result of a previous SRDF/Metro configuration:

- Devices must be unmapped and unmasked.
- Devices should not be SRDF devices.
- Device must have a federated WWN.
- Device should not be Data Domain devices.

The following SYMCLI commands have been enhanced with a new set option – `no_identity` to support restoring the personality of devices removed from SRDF/Metro configurations:

- Devices: `symdev set -no_identity`
- Device groups: `symdg set -no_identity`
- Composite groups: `symcg set -no_identity`
- Storage groups: `symsg set -no_identity`

The steps to restore device personality vary depending on whether the bias was changed before the devices are deleted from the SRDF/Metro group configuration.

If the bias was *not* changed before the `deletepair` operation, perform only Steps 1 - 2 below.

If bias was changed before the `deletepair` operation:

- The R1 (the original R2) will have the original R1's ID
- The R2 (the original R1) will have the original R2's ID.

Both sides of the device pair may need to be replaced. Use the `symdev show` command to display which IDs need to be reset.

#### Procedure

1. Use the `half_deletepair` or `deletepair` operation to remove all devices from the SRDF/Metro group configuration.
2. Use the applicable `set -no_identity` command to restore the native identity of the specified device, or all the devices in the specified group.

To restore the personality of R2 (now non-SRDF) devices in storage group RDF\_2SG:

```
symsg -sid 248 -sg RDF2_SG set -no_identity
```

## Display SRDF/Metro

In addition to the examples in the previous section, this section describes changes to the output of `show` and `list` commands to display devices in SRDF/Metro configurations.

#### **symdev show**

Output of the `symdev show` command displays the ActiveActive or ActiveBias pair state. Changes include:

- New RDF pair state (RDF Pair State ( R1 <===> R2 ) of ActiveActive or ActiveBias



- New RDF mode of Active for an SRDF device

The following output is for an R1 device when it is in an SRDF/Metro configuration and the pair state is ActiveActive:

```
symdev show 3F -sid 085
```

```

Device Physical Name      : /dev/sdam
Device Symmetrix Name     : 0003F
Device Serial ID         : 850003F000
Symmetrix ID             : 000197100085
. . .
Device Service State      : Normal

Device Status             : Ready              (RW)
Device SA Status          : Ready              (RW)
Device User Pinned        : False
Host Access Mode          : Active
Device Tag(s)             : None
. . .
RDF Information
{
  Device Symmetrix Name    : 0003F
  RDF Type                : R1
  RDF (RA) Group Number   : 86 (55)

  Remote Device Symmetrix Name : 0008E
  Remote Symmetrix ID       : 000197100086
  . . .
  RDF Mode                : Active
  RDF Adaptive Copy        : Disabled
  RDF Adaptive Copy Write Pending State : N/A
  RDF Adaptive Copy Skew (Tracks) : 65535
  . . .
  Device Suspend State     : N/A
  Device Consistency State : Enabled
  Device Consistency Exempt State : Disabled
  RDF R2 Not Ready If Invalid : Disabled
  . . .
  Device RDF State         : Ready              (RW)
  Remote Device RDF State : Ready              (RW)
  RDF Pair State ( R1 <==> R2 ) : ActiveActive
  . . .

```

The following output is for an R2 device when it is in an SRDF/Metro configuration and the pair state is ActiveActive:

```
symdev show 8E -sid 086
```

```

Device Physical Name      : /dev/sdac
Device Symmetrix Name     : 0008E
Device Serial ID         : 85000C8000
Symmetrix ID             : 000197100086
. . .
Device Service State      : Normal

Device Status             : Ready              (RW)
Device SA Status          : Ready              (RW)
Device User Pinned        : False
Host Access Mode          : Active
Device Tag(s)             : None
. . .
RDF Information
{
  Device Symmetrix Name    : 0008E

```

```

RDF Type : R2
RDF (RA) Group Number : 85 (54)

Remote Device Symmetrix Name : 0003F
Remote Symmetrix ID : 000197100085
. . .
RDF Mode : Active
RDF Adaptive Copy : Disabled
RDF Adaptive Copy Write Pending State : N/A
RDF Adaptive Copy Skew (Tracks) : 65535
. . .
Device Suspend State : N/A
Device Consistency State : Enabled
Device Consistency Exempt State : Disabled
RDF R2 Not Ready If Invalid : Disabled
. . .
Device RDF State : Ready (RW)
Remote Device RDF State : Ready (RW)
RDF Pair State ( R1 <==> R2 ) : ActiveActive
. . .

```

### symcfg list -rdfg

Changes to output of the symcfg list -rdfg command include:

- Whether or not the SRDF group is online (Group (S)tatus = O).
- Whether or not an SRDF group is a Witness SRDF group (Group (T)ype = W).
- Whether or not the device pairs in the SRDF group are configured for SRDF/Metro (Group Flag M = X).

### symcfg list -rdfg all -sid 084

Symmetrix ID : 000197100084

#### S Y M M E T R I X R D F G R O U P S

| Local    |        | Remote   |              | Group |         |                 | RDFA Info |            |                |
|----------|--------|----------|--------------|-------|---------|-----------------|-----------|------------|----------------|
| RA-Grp   | LL sec | RA-Grp   | SymmID       | ST    | Name    | Flags LPDS CHTM | Dir Cfg   | Flags CSRM | Cycle time Pri |
| 115 (72) | 10     | 116 (73) | 000197100086 | OD    | sdp_dg3 | XX.. ..XX       | F-S       | -IS-       | 15 33          |
| 120 (77) | 10     | 117 (74) | 000197100086 | OW    | sdp_dg4 | XX.. ..X.       | F-S       | -IS-       | 15 33          |
| 121 (78) | 10     | 118 (75) | 000197100086 | FD    | sdp_dg5 | XX.. ..X.       | F-S       | -IS-       | 15 33          |

Legend:

```

Group (S)tatus : O = Online, F = Offline
Group (T)ype : S = Static, D = Dynamic, W = Witness
Director (C)onfig : F-S = Fibre-Switched, F-H = Fibre-Hub
                     G = GIGE, E = ESCON, T = T3, - = N/A
Group Flags :
  Prevent Auto (L)ink Recovery : X = Enabled, . = Disabled
  Prevent RAs Online Upon (P)ower On: X = Enabled, . = Disabled
  Link (D)omino : X = Enabled, . = Disabled
  (S)TAR/SQAR mode : N = Normal, R = Recovery, . = OFF
                    S = SQAR Normal, Q = SQAR Recovery
  RDF Software (C)ompression : X = Enabled, . = Disabled, - = N/A
  RDF (H)ardware Compression : X = Enabled, . = Disabled, - = N/A
  RDF Single Round (T)rip : X = Enabled, . = Disabled, - = N/A
RDF (M)etro : X = Configured, . = Not Configured
RDFA Flags :
  (C)onsistency : X = Enabled, . = Disabled, - = N/A
  (S)tatus : A = Active, I = Inactive, - = N/A
  (R)DFA Mode : S = Single-session, M = MSC, - = N/A
  (M)sc Cleanup : C = MSC Cleanup required, - = N/A

```

**symcfg list -rdfg -rdf\_metro**

The `-rdf_metro` option shows information specific to SRDF/Metro. Changes include:

- RDFA Info displayed in the default output (above) is replaced with information specific to SRDF/Metro.
- Whether the SRDF group was enabled for Witness or bias protection during the establish/restore.
- Whether Witness or bias protection is currently in effect.
- SRDF groups that have Witness protection in effect identify the Witness array.

In the following example,

- Group 115 on array 000197100084:
  - Contains SRDF device pairs that are configured for SRDF/Metro;
  - Is configured to use Witness protection;
  - Is currently Witness-protected; and
  - The Witness array is 000197100087.
- Group 116 on array 000197100084
  - Contains SRDF device pairs that are configured for SRDF/Metro;
  - Is configured to use Witness protection; but
  - Is currently using bias.
- Group 117 on Symmetrix 000197100084:
  - Contains SRDF device pairs that are configured for SRDF/Metro;
  - Is configured to use bias; and
  - Is currently using bias.
- Group 125 on Symmetrix 000197100084:
  - Contains devices that are configured for SRDF/Metro;
  - Is configured to use Witness protection; but
  - Its Witness protecton is degraded (only one side can see the Witness array);
  - The Witness array is 000197100087

```
symcfg list -rdfg all -sid 084 -rdf_metro
```

```
Symmetrix ID : 000197100084
```

### S Y M M E T R I X   R D F   G R O U P S

| Local    |        | Remote   |              | Group     |          |                 | RDF Metro |              |                |
|----------|--------|----------|--------------|-----------|----------|-----------------|-----------|--------------|----------------|
| RA-Grp   | LL sec | RA-Grp   | SymmID       | ST        | Name     | Flags LPDS CHTM | Dir Cfg   | Witness CE S | SymmID         |
| 115 (72) | 10     | 116 (73) | 000197100086 | OD        | sdp_dg3  | XX.. ..XX       | F-S       | <b>WW</b>    | N 000197100087 |
| 125 (7C) | 10     | 126 (7D) | 000197100086 | OD        | sdp_dg13 | XX.. ..XX       | F-S       | <b>WW</b>    | D 000197100087 |
| 120 (77) | 10     | 117 (74) | 000197100087 | <b>OW</b> | sdp_dg4  | XX.. ..X.       | F-S       | --           | - -            |
| 121 (78) | 10     | 118 (75) | 000197100086 | FD        | sdp_dg5  | XX.. ..X.       | F-S       | --           | - -            |
| 116 (73) | 10     | 119 (76) | 000197100086 | OD        | sdp_dg7  | XX.. ..XX       | F-S       | WB           | F -            |
| 117 (74) | 10     | 120 (77) | 000197100086 | OD        | sdp_dg9  | XX.. ..XX       | F-S       | BB           | - -            |

**Legend:**

Group (S)tatus : O = Online, F = Offline  
 Group (T)ype : S = Static, D = Dynamic, Q = Witness

```

Director (C)onfig : F-S = Fibre-Switched, F-H = Fibre-Hub
                   G = GIGE, E = ESCON, T = T3, - = N/A

Group Flags      :
  Prevent Auto (L)ink Recovery      : X = Enabled, . = Disabled
  Prevent RAs Online Upon (P)ower On: X = Enabled, . = Disabled
  Link (D)omino                     : X = Enabled, . = Disabled
  (S)TAR/SQAR mode                  : N = Normal, R = Recovery, . = OFF,
                                     S = SQAR Normal, Q = SQAR Recovery
  RDF Software (C)ompression         : X = Enabled, . = Disabled, - = N/A
  RDF (H)ardware Compression         : X = Enabled, . = Disabled, - = N/A
  RDF Single Round (T)rip            : X = Enabled, . = Disabled, - = N/A
  RDF (M)etro                       : X = Configured, . = Not Configured

RDF Metro Flags :
  (C)onfigured Type                 : W = Witness, B = Bias, - = N/A
  (E)ffective Type                  : W = Witness, B = Bias, - = N/A
  Witness (S)tatus                  : N = Normal, D = Degraded,
                                     F = Failed, - = N/A

```

# CHAPTER 6

## Consistency Group Operations

This chapter describes the following topics:

- [Consistency group operations overview](#) ..... 182
- [SRDF consistency group operations](#) ..... 184
- [Enable and disable SRDF consistency protection](#) ..... 188
- [Modify consistency groups](#) ..... 197
- [Consistency groups with a parallel database](#) ..... 207
- [Consistency groups with BCV access at the target site](#) ..... 208

## Consistency group operations 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 rapid recovery 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 (storrd) provides consistency protection for:

- SRDF/A Multi-Session Consistency (MSC) consistency groups in multi-array environments
- SRDF/S RDF-Engenuity 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, storrd:
  - 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, storrd continuously polls SRDF/S sessions for data flow interruptions.  
If any R1 device is unable to propagate data to its R2 target, storrd:
  - Halts all R1->R2 data flow within an RDF-ECA consistency group.

storrd 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 consistency group operations

Before storrd 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).

## Enable 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.

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.

### Syntax

Use the following SYMAPI options file setting to enable storrdfd:

```
SYMAPI_USE_RDFD=ENABLE
```

## Enable 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, manually recreate the updated CG definition on each one of these hosts.

### NOTICE

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*

### Syntax

Enable GNS on each host using the following SYMAPI options file setting:

```
SYMAPI_USE_GNS=ENABLE
```

## Redundant consistency protection

Two instances of the SRDF daemon can run simultaneously on separate control hosts to create redundant consistency protection for composite groups.

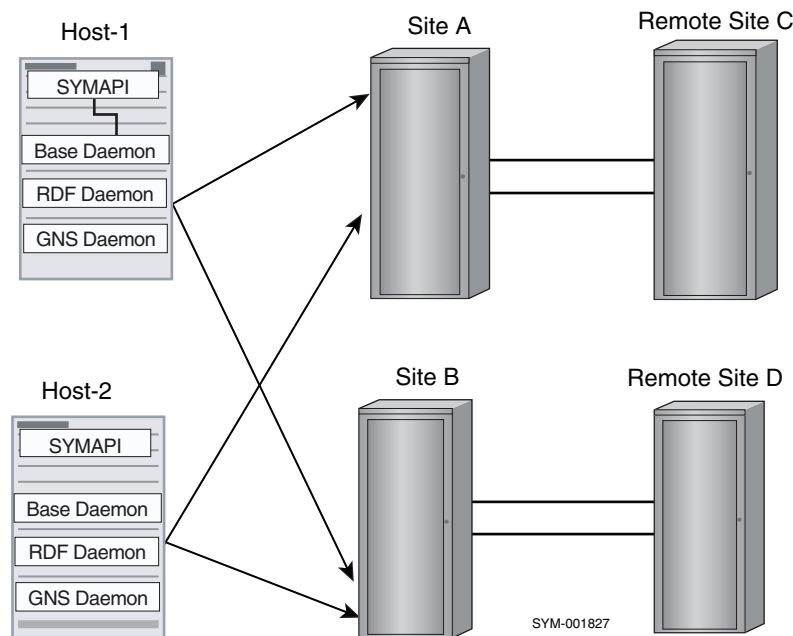
Simultaneous SRDF daemons perform independent monitoring and switching operations. If one fails, the other SRDF daemon takes it place, and completes all pending tasks (commit the last cycle to the target site).

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 the following image, or
- Manually define the composite group on all control hosts.

**Figure 18** Running redundant hosts to ensure consistency protection

In the image above, Host-1 and Host-2 run all three daemons: base daemon, SRDF daemon, and GNS daemon to ensure data consistency protection

#### NOTICE

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 operations and applicable pair states on page 438](#) 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 a 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.

**Procedure**

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 ConsistGrp 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 ConsistGrp 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 `ConsistGrp` CG.

## Create composite groups from various sources

Sources from which to create a composite group include:

- Device group - Translate the devices of an existing device group
- RDMS database - Translate the devices of an existing RDBMS database or tablespace
- Volume group - Translate the devices of an existing logical volume group

---

### Note

The E-Lab™ Interoperability Navigator at <http://elabnavigator.EMC.com> provides detailed interoperability information.

---

## Create a composite group from an existing device group

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.

### Example

In the following example, the `symdg` command:

- Translates devices to SRDF
- Adds all devices from a device group `Symm64DevGrp` to a composite group `ConsistGrp`.
- Adds the composite group to the SRDF consistency database on the host
- Enables the group for SRDF consistency protection:

```
symdg dg2cg Symm64DevGrp ConsistGrp -rdf_consistency
```

## Create a composite group from an RDBMS database

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.

---

**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, 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.

An Oracle connection string such as the Transparent Network Substrate (TNS) is required.

**Examples**

In the following example, a local connect is used. The `export` command sets the variable to a username of "array" and a password of "manager".

```
export SYMCLI_RDB_CONNECT=array/manager
```

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, add a string to indicate the ODBC data source administrator.

To add string "HR":

```
set SYMCLI_RDB_CONNECT=array/manager@HR
```

Optionally, 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.

To set the environmental variable to `oracle` :

```
export SYMCLI_RDB_TYPE=oracle
```

**Translate devices in a composite group**

You can translate the devices in a database to a composite group.

You can translate the devices in an Oracle type tablespace to a composite group.

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`.

**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
```

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
```

## Create a composite group from a logical volume group

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
```

## Enable and disable 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). 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.
- 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. [Modify consistency groups on page 197](#) provides more information.

## Enable consistency: composite group vs. SRDF group name

Consistency protection can be enabled and disabled at the composite group level or at the SRDF group name level:

- When consistency is enabled at the composite group level, all devices within the consistency group operate as a single unit.
- When consistency protection is enabled at the SRDF group name level, only the devices in the specified SRDF group operate as a unit.

### Enable/disable consistency for a composite group

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
```

### Enable consistency for an SRDF group

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. [Modify consistency groups on page 197](#) 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.

### Examples

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.

To create two named subsets of the composite group:

One containing the asynchronous SRDF groups:

```
symcpg -cg SALES set -name sales1 -rdfg 76:100
symcpg -cg SALES set -name sales1 -rdfg 77:101
```

One containing the synchronous SRDF groups:

```
symcpg -cg SALES set -name sales2 -rdfg 76:120
symcpg -cg SALES set -name sales2 -rdfg 77:121
```

To enable independent consistency protection for the two subsets:

```
symcpg -cg SALES enable -rdfg name:sales1
symcpg -cg SALES enable -rdfg name:sales2
```

## Enable/disable consistency protection for 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
```

## Enable/disable consistency protection for 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:

- When consistency is enabled for concurrent devices at the composite group level, all device mirrors must be operating in the same SRDF mode; that is all device mirrors must be operating either synchronously or asynchronously.
- When consistency is enabled for concurrent devices at the SRDF group name level, the SRDF daemon monitors the SRDF groups separately.

## Enable/disable consistency for concurrent devices in a composite group

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.

### Syntax

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
```

## Enable consistency for concurrent devices in a SRDF group

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.
- The following table lists the combinations of consistency protection modes allowed for the mirrors of a concurrent relationship.

**Table 16** Consistency modes for concurrent mirrors

| R1→R2 (first mirror) | R1→R2 (second mirror) |
|----------------------|-----------------------|
| MSC                  | None                  |
| MSC                  | RDF-ECA               |



**Table 16** Consistency modes for concurrent mirrors (continued)

| R1→R2 (first mirror) | R1→R2 (second mirror) |
|----------------------|-----------------------|
| MSC                  | MSC                   |
| RDF-ECA              | None                  |
| RDF-ECA              | RDF-ECA               |
| RDF-ECA              | MSC                   |
| None                 | None                  |
| None                 | MSC                   |
| None                 | RDF-ECA               |

## Enabling consistency for concurrent pairs

### Procedure

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.

## Check if device pairs are enabled for consistency protection

### Syntax

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

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.

**Examples**

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'.
```

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
```

**Block 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.

**Delete an SRDF consistency group**

When you delete an SRDF consistency group from a CG, the SRDF daemon stops monitoring the CG.

**NOTICE**

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. [Enable and disable SRDF consistency protection on page 188](#) provides more information.

**Syntax**

```
symcg delete GroupName
```

**Options****-force**

Required if the group is disabled and there are members in the group.

**-symforce**

Required if the group is enabled. The composite group remains enabled but is removed from the SYMAPI database.

**Example**

To delete a disabled SRDF consistency group `mycg1` (with members):

```
symcg delete mycg1 -force
```

## Suspend SRDF consistency protection

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.

Consistency protection is automatically restored upon resumption of the link.

Consistency protection is not disabled unless you specify `symcg -cg disable`.

**Syntax**

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.

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.

The state of the R2 devices at the end of the deactivation varies depending on whether the `suspend` or `split` command is used:

**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.

**Options**

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.

**Note**

The `-force` option is required.

**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
```

**Verify SRDF consistency****Examples**

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
```

**Composite group cleanup (msc\_cleanup)**

When an SRDF/A single mode session is dropped, the OS 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.

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.

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.

**Syntax**

Use the `msc_cleanup` command to cleanup after a session is dropped for devices operating in SRDF/A mode with consistency enabled MSC. 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 to 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.

**Examples**

To cleanup a composite group (mycg):

```
symrdf -cg mycg msc_cleanup
```

To cleanup from the remote host at the R2 site for array 123 and direct the command to SRDF group 4:

```
symrdf -sid 123 -rdfg 4 msc_cleanup
```

## Modify 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 consistency group modification

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.

**NOTICE**

EMC strongly recommends running GNS on your hosts to ensure consistency protection while dynamically modifying CGs.

## Consistency group modification restrictions

The following apply to dynamic `add` and `remove` options of the `symcg modify` command:

- Devices that are in an SRDF/Metro configuration cannot be added to SRDF CGs
- A CG that contains devices that are in an SRDF/Metro configuration cannot be enabled for SRDF consistency.
- Engenuity 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.
- 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.

---

## Prepare staging area for consistency group modification

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.
- Staging area cannot be an SRDF/Metro 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

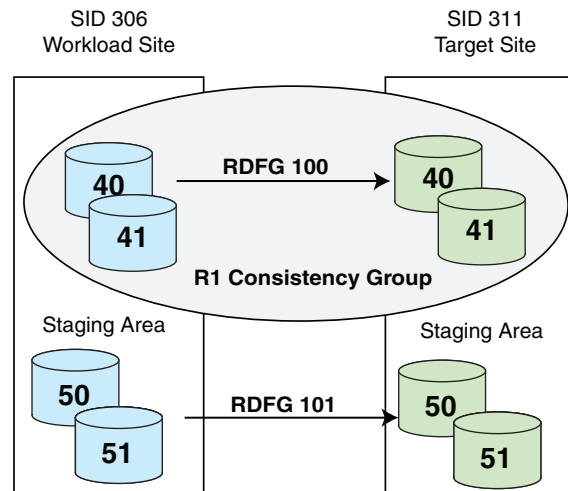
The dynamic modify add operation moves device pairs from the staging area into the SRDF groups of a consistency group.

All devices in the staging area must be in one of the following SRDF pair states for each SRDF group:

- Synchronized
- SyncInProgress with invalid tracks owed to the R2
- Suspended
- Suspended with invalid tracks owed to the R2

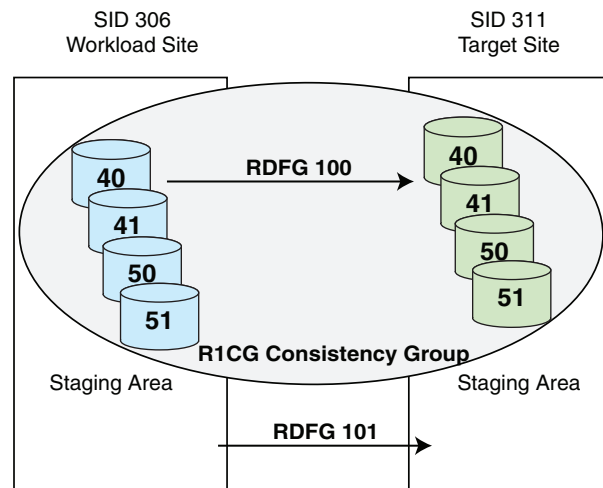
If any device pair is Suspended (with or without invalid tracks on any of its SRDF groups), then the device pairs in the same SRDF group must all be Suspended.

The following image shows a staging area for an R1-R2 configuration:

**Figure 19** 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.

The following image shows the R1CG consistency group after the dynamic add operation:

**Figure 20** 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.

## Prepare the staging area to remove devices

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.

---

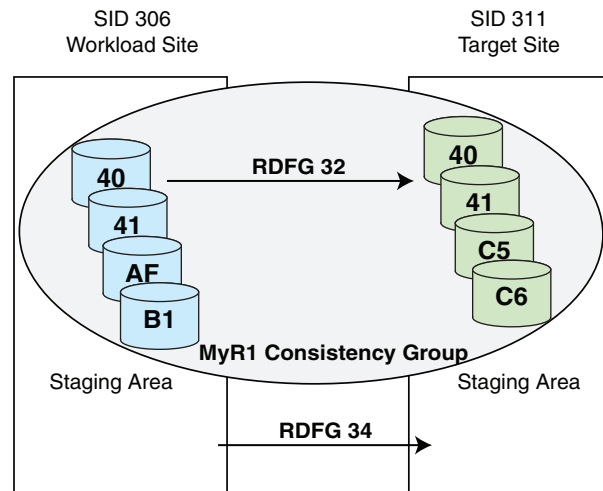
### Note

The dynamic modify remove operation must never leave an SRDF group empty.

---

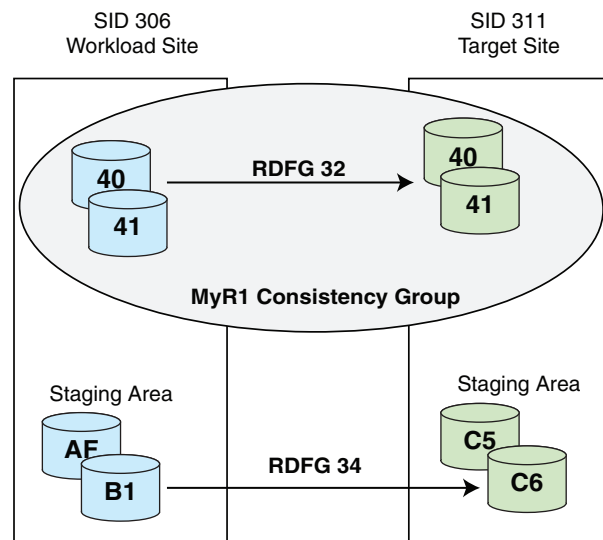
The following image shows empty group RDFG 34 configured to receive devices removed from RDFG 32:



**Figure 21** 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.

The following image shows the MyR1 consistency group and its staging area after the dynamic modify remove operation has completed.

**Figure 22** MyR1 CG after a dynamic modify remove operation

## Restrictions: Add devices to SRDF consistency group

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 devices from SRDF consistency group

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.

## Restrictions: Device types allowed for add operations to an RDF1 consistency group

The following table 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 17** 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
```

## Restrictions: Device types and consistency modes allowed for add operations to a concurrent RDF1 consistency group

Before you perform this procedure, review [Enabling SRDF consistency protection for concurrent SRDF devices on page 191](#).

The following table lists 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 18** 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                                                       |

The following table lists the allowable consistency modes for the SRDF groups of a concurrent CG.

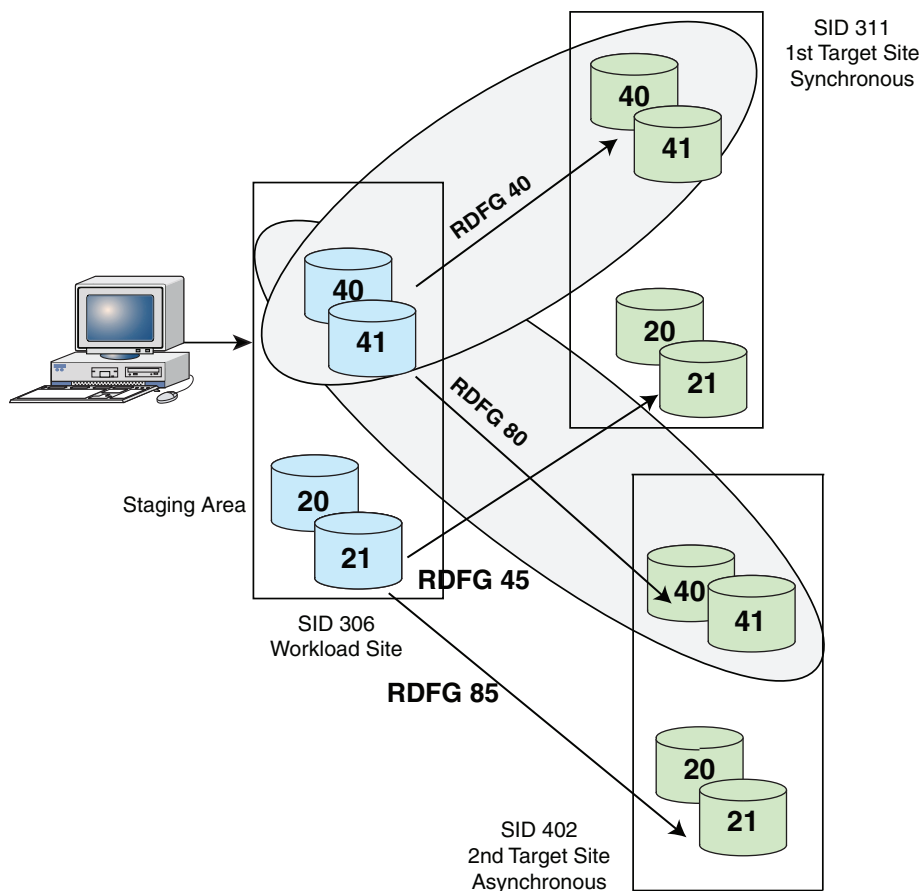
**Table 19** Supported consistency modes for concurrent SRDF groups

| SRDF group 1 (first mirror) | SRDF group 2 (second mirror) |
|-----------------------------|------------------------------|
| RDF-ECA                     | RDF-ECA                      |
| RDF-ECA                     | MSC                          |
| RDF-ECA                     | Not enabled                  |
| Not enabled                 | RDF-ECA                      |
| MSC                         | RDF-ECA                      |
| MSC                         | MSC                          |
| MSC                         | Not enabled                  |
| Not enabled                 | MSC                          |

### Examples

In this example, device 20 is added to two independently-enabled SRDF groups of a CG.

The following image shows the staging area shared by array 306, 311, and 402 in a concurrent SRDF configuration:

**Figure 23** Adding a device to independently-enabled SRDF groups of a concurrent CG

The staging area contains devices 20 and 21.

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`:

```
symcgs -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
```

## Restrictions: Devices types allowed to add to a cascaded RDF1 consistency group

Before you perform this procedure, review [Check if device pairs are enabled for consistency protection on page 193](#).

The following table lists 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 20** 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<br>Hop 2 not enabled | Hop 1 enabled<br>Hop 2 enabled | Hop 1 not enabled<br>Hop 2 enabled | Hop 1 enabled<br>Hop 2 not enabled | Hop 1 enabled<br>Hop 2 enabled | Hop 1 not enabled<br>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                        |

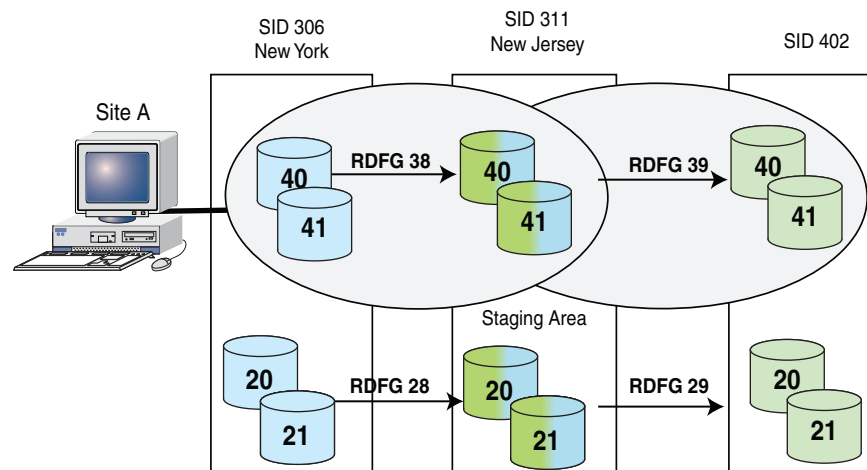
The following table lists the allowable consistency modes for the hops of a cascaded CG.

**Table 21** Supported consistency modes for cascaded hops

| R1→R21 (hop 1) | R21→R2 (hop 2) |
|----------------|----------------|
| RDF-ECA        | MSC            |
| RDF-ECA        | Not enabled    |
| MSC            | Not enabled    |

### Examples

The following image shows a cascaded SRDF configuration sharing the staging area among array 306, 311, and 402:

**Figure 24** 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:

```
symcgs -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
```

## Restrictions: Device types allowed for remove operations from an RDF1 consistency group

The following table lists 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 22** 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:

```
symcgs -cg R1CG modify -remove -sid 306 -stg_rdfg 101 -devs 50:51 -cg_rdfg 100
```

## Restrictions: Device types allowed for remove operations from a concurrent RDF1 consistency group

The following table lists 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 23** 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:

```
symcgs -cg ConCG modify -remove -sid 306 -stg_rdfg 71,81 -devs 20:30 -cg_rdfg 70,80
```

## Restrictions: Device types allowed for remove operations from a cascaded RDF1 consistency group

The following table lists 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 24** 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   |                                |                                    |
|--------------------|------------------------------------|--------------------------------|------------------------------------|------------------------------------|--------------------------------|------------------------------------|
|                    | Hop 1 enabled<br>Hop 2 not enabled | Hop 1 enabled<br>Hop 2 enabled | Hop 1 not enabled<br>Hop 2 enabled | Hop 1 enabled<br>Hop 2 not enabled | Hop 1 enabled<br>Hop 2 enabled | Hop 1 not enabled<br>Hop 2 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

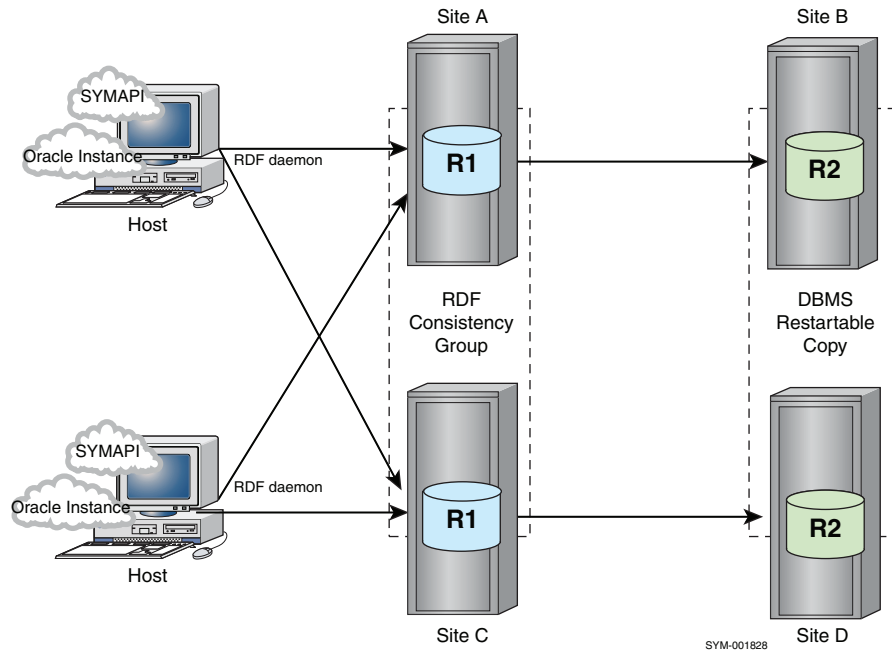
For example, if a concurrent R11 loses a link to one of its mirrors during a dynamic modify add operation, the recover operation may remove all devices added to the CG by this operation. This ensures that the CG device pairs are consistent at all three sites.

## Consistency groups with a parallel database

The following images shows 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 25** 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.

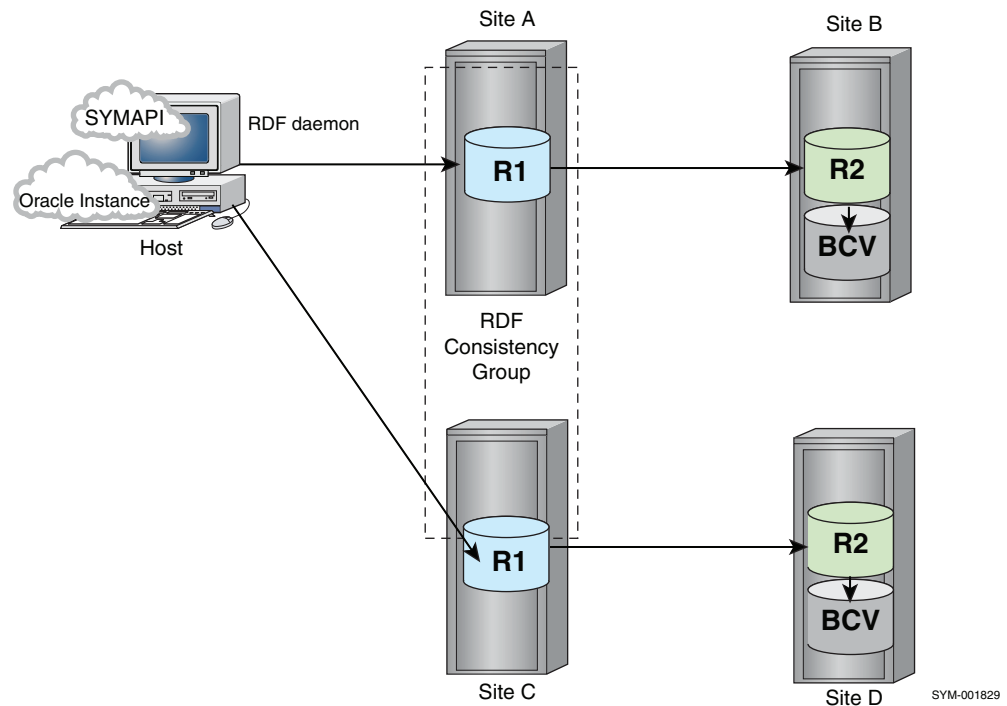
[Figure 25 on page 208](#) shows the SRDF daemons located on the production hosts. EMC recommends that you do not run the SRDF daemon on the same control host running database applications.

## 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.

The following image shows a configuration with target-side BCVs that mirror the R2 devices:



**Figure 26** 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 207](#) 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 26 on page 209](#) shows the SRDF daemons located on the production hosts.

---

#### Note

EMC recommends: Do not run the SRDF daemon on the same control host running database applications.

---



# CHAPTER 7

## Concurrent Operations

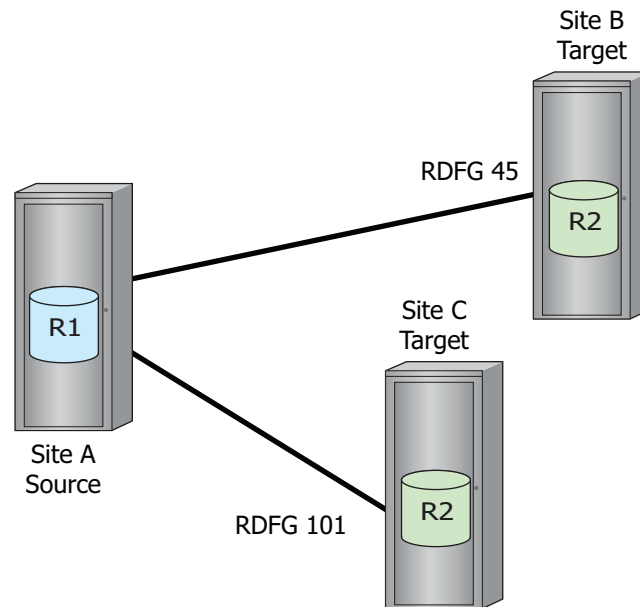
This chapter describes the following topics:

- [Concurrent operations overview](#)..... 212
- [Configuring a concurrent SRDF relationship](#)..... 214

## Concurrent operations overview

In a concurrent SRDF configuration, the source R1 device is mirrored to two R2 devices on two different remote arrays.

**Figure 27** Concurrent SRDF



The two R2 devices operate independently but concurrently using any combination of SRDF modes.

---

### Note

Starting with Engenuity 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.

## Concurrent operations restrictions

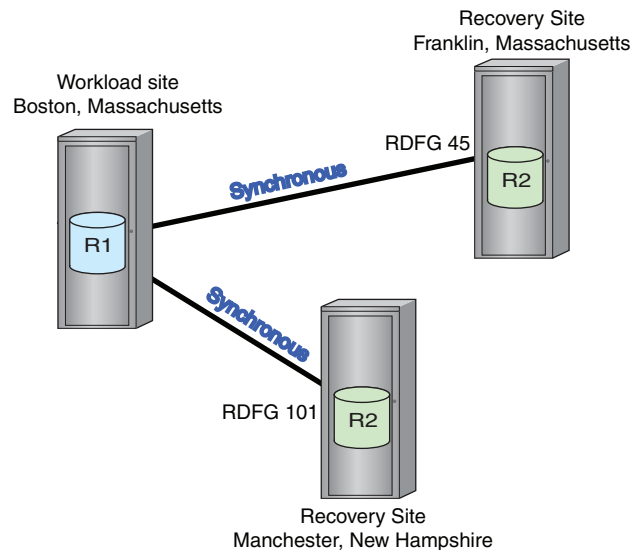
- Devices that are part of an SRDF/Metro configuration cannot be mirrored to two remote devices, so cannot be part of a concurrent configuration.
- The R2 devices at each remote array must belong to a different SRDF group.
- Asynchronous SRDF to both R2 devices requires Engenuity 5875 or higher on the R1 side.
- Simultaneous restore from both R2 devices to the R1 device cannot be performed.
- Both mirrors of an SRDF device cannot be swapped at the same time.

### Restrictions: 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.

The following image shows three sites that are within synchronous distance:

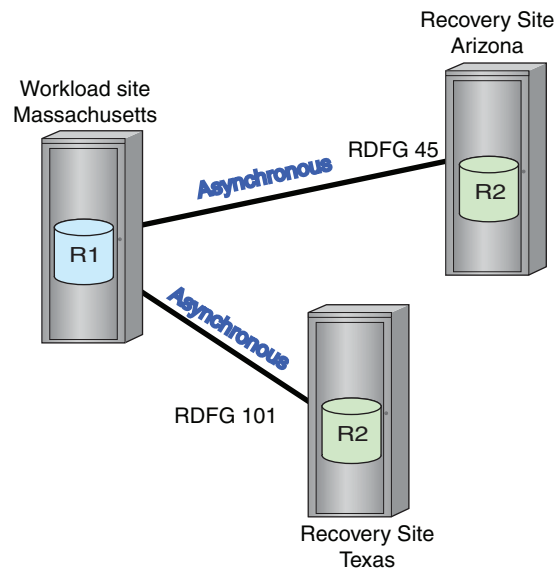
**Figure 28** Concurrent SRDF/S to both R2 devices



### Restrictions: 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 29** 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.

The 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.

## Additional documentation for concurrent operations

### 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 operations and applicable pair states on page 432](#) provides more information.

### Consistency protection

You can enable consistency protection for devices in a concurrent configuration.

[Enable consistency for concurrent devices in a SRDF group on page 192](#) provides more information.

---

#### Note

Consistency protection for two mirrors participating in different consistency groups with SRDF/S requires Engenuity 5874 and higher.

---

## Configuring a concurrent SRDF relationship

To configure a concurrent SRDF relationship:

### Procedure

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 and establishing concurrent SRDF devices

To create a device group for the concurrent SRDF devices and initially synchronize (establish) the devices across the concurrent SRDF links:

### Procedure

1. Use the `symdbg` command to create an R1 device group.

```
symdbg [-i Interval] [-c Count] [-v]
.....
create DgName -type RDF1
```

```
symdbg create ConcGrp -type RDF1
```

2. Use the `symdbg add` command to add all concurrent SRDF devices to the device group:

```
symdbg -g DgName [-i Interval] [-c Count] [-v]
```

```
....
add dev SymDevName
```

```
symdbg add dev 0001 -g ConcGrp -sid 0001
```

```
symdbg add dev 0021 -g ConcGrp
```

```
symdbg 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
```

Alternatively, 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 Continuation Volume (BCV) devices cannot contain concurrent SRDF mirrors.

---

## Split concurrent SRDF devices

### Syntax

Use the `symrdf split` command to split concurrent SRDF pairs, either one at a time or at the same time.

---

#### Note

Applicable only to Engenuity 5875 and higher: Concurrent R1 devices can have two mirrors participating in different consistency groups with MSC consistency protection enabled.

---

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
```

To split the concurrent pairs simultaneously:

```
symrdf -g DgName split -rdfg All
```

### Examples

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 for device group concGrp at the same time:

```
symrdf -g concGrp split -rdfg ALL
```

## Restore concurrent devices

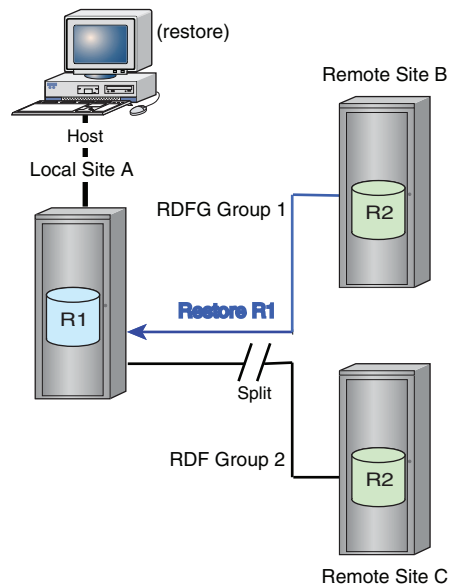
In concurrent configuration, there are two RDFG groups of R2 devices.

- 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.

### Restore R1 from a concurrent R2

Use the `restore` command to restore only the R1 device from the specified R2:

**Figure 30** 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.



## Syntax

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
```

To re-establish the R2 devices not used in the restore operation:

```
symrdf -g DgName restore -rdfg GroupNum of group not used to restore
```

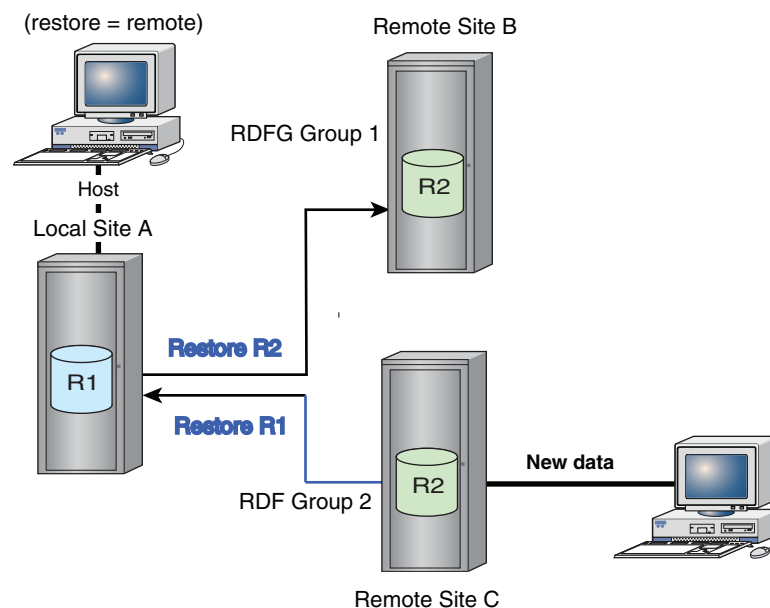
To re-establish second mirror (RDFG 2) for group `concGrp`:

```
symrdf -g concGrp establish -rdfg 2
```

## Restore 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 31** 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.

---

**Syntax**

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
```

**Examples**

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
```

**View 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

|            |       | RDF<br>Typ:G | STATUS |    |     | MODES |                  | RDF S T A T E S  |     |                 |
|------------|-------|--------------|--------|----|-----|-------|------------------|------------------|-----|-----------------|
| Sym<br>Dev | RDev  |              | SA     | RA | LNK | MDATE | R1 Inv<br>Tracks | R2 Inv<br>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 |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |
|                  |  |  |  |  |                  |  |  |  |  |       |  |

|        |       |    |   |       |    |      |    |   |   |      |            |
|--------|-------|----|---|-------|----|------|----|---|---|------|------------|
|        |       | 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 8

## Cascaded Operations

This chapter describes the following topics:

- [Cascaded operations overview](#) ..... 222
- [Setting up cascaded SRDF](#) ..... 225
- [R21 device management](#) ..... 228
- [Cascaded SRDF with EDP](#) ..... 230
- [Sample session: planned failover](#) ..... 236
- [Display cascaded SRDF](#) ..... 238

## Cascaded operations overview

Cascaded SRDF is a three-way data mirroring and recovery solution that consists of:

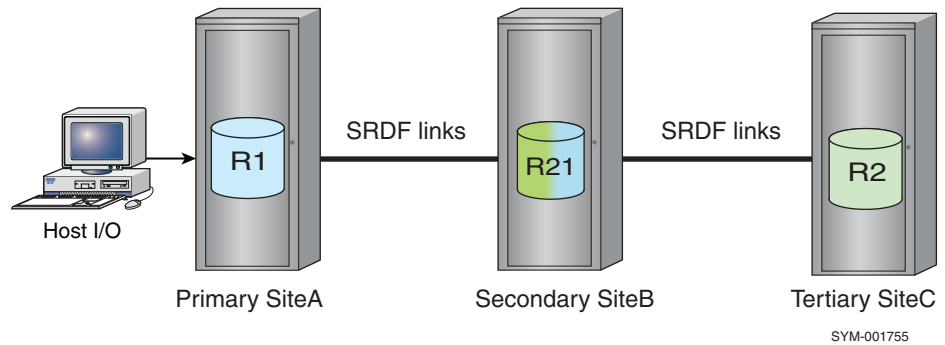
- A R1 device replicating data to
- An R21 device at a secondary site, which replicates the same data to a
- R2 device located at a tertiary site

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.

The following image shows a basic cascaded SRDF configuration.

**Figure 32** 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.

The following tables list the applicable pair states for cascaded operations:

- [Cascaded SRDF control operations and applicable pair states on page 424](#)
- [Cascaded SRDF set operations and applicable pair states on page 430](#)

**Note**

To perform cascaded SRDF operations with Access Control enabled, you need SRDF BASECTRL, BASE, and BCV access types. *EMC Solutions Enabler Symmetrix Array Management CLI Product 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).

[SRDF modes in cascaded configurations with EDP on page 224](#) lists the SRDF modes supported from R1 -> R21, and R21 -> R2 when EDP is configured and the R21 device is diskless.

The following table lists the SRDF modes supported from R1 -> R21, and R21 -> R2 when the R21 device is NOT diskless.

**Table 25** SRDF modes for cascaded configurations (no EDP)

| R1 -> R21                        | R21 -> R2                          |
|----------------------------------|------------------------------------|
| Adaptive copy disk               | Asynchronous<br>Adaptive copy disk |
| Adaptive copy write pending*     | Asynchronous<br>Adaptive copy disk |
| Asynchronous (no EDP)            | Asynchronous<br>Adaptive copy disk |
| Semi-synchronous (for 5671 only) | Asynchronous<br>Adaptive copy disk |
| Synchronous                      | Asynchronous<br>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 26** SRDF modes for cascaded configurations with EDP

| R1 -> Diskless R21                                                | Diskless R21 -> R2           |
|-------------------------------------------------------------------|------------------------------|
| Synchronous<br>Adaptive copy disk<br>Adaptive copy write pending* | Asynchronous                 |
| Synchronous<br>Adaptive copy disk<br>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 operations

- Devices that are part of an SRDF/Metro configuration cannot at the same time be part of a cascaded configuration.
- 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 relationship for cascaded SRDF

Setting up a cascaded SRDF relationship is a two-step process:

#### Procedure

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.

---

### Create cascaded SRDF pairs and set mode

#### Syntax (-file option)

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 -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

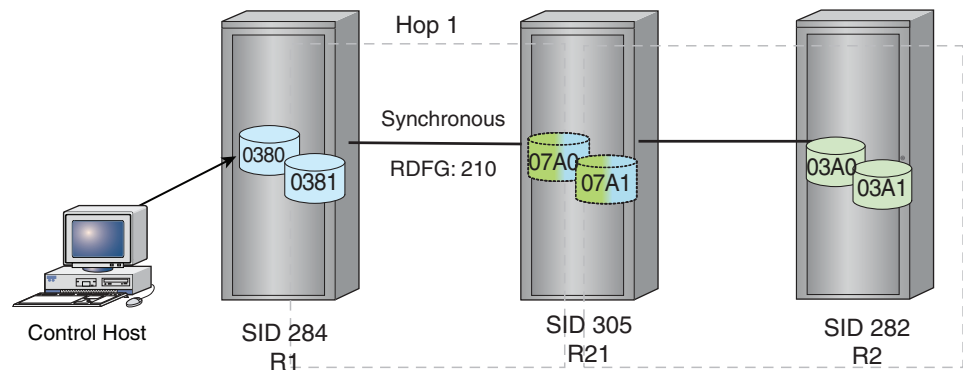
In the following example, TestFile1 specifies two device pairs on SIDs 284 and 305:

```
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 33** Configuring the first hop



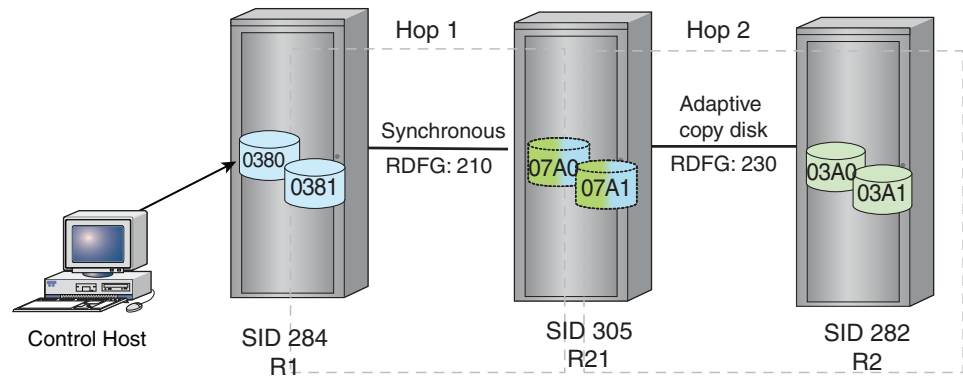
The SRDF R1 → R2 device pairs are created and established in SRDF synchronous mode.

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 34** 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.

The following tables list the applicable pair states for cascaded operations:

- [Cascaded SRDF control operations and applicable pair states on page 424](#)
- [Cascaded SRDF set operations and applicable pair states on page 430](#)

## 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.

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:

1. To create an empty RDF1 composite group testcg:

```
symcg -type rdf1 create testcg
```

2. 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
```

3. 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
```

4. To set the SRDF group name to name1:

```
symcg -cg testcg set -name name1 -rdfig 284:210,256:60
```

## R21 device management

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.

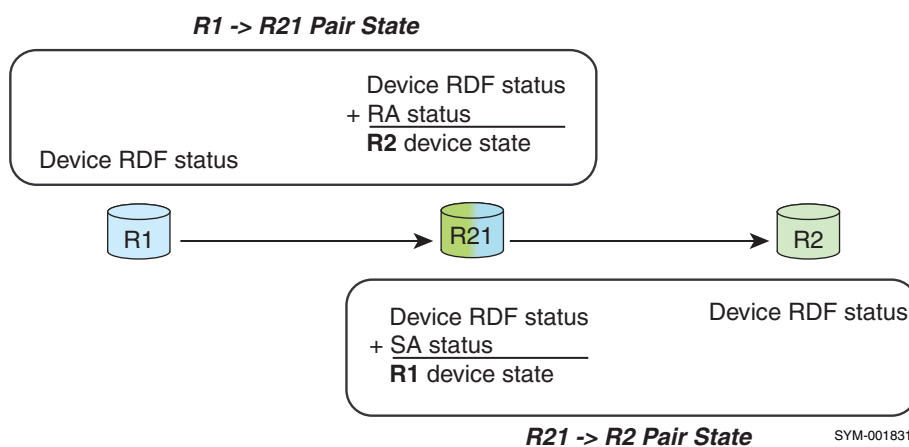
When controlling an R2 device in a cascaded SRDF relationship, the first hop represents the R2->R21 relationship and the second hop represents the R21-> R1 relationship.

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.

The following image shows how the R21 SRDF device state is determined and how each SRDF mirrored pair state is determined.

**Figure 35** Determining SRDF pair state in cascaded configurations



Device actions modify only the SA status of the R21 device.

For example, if `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 `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 424](#) provides more information.

## Hop 2 controls in cascaded SRDF

You can perform control operations from hosts connected any of the three arrays in a cascaded configuration.

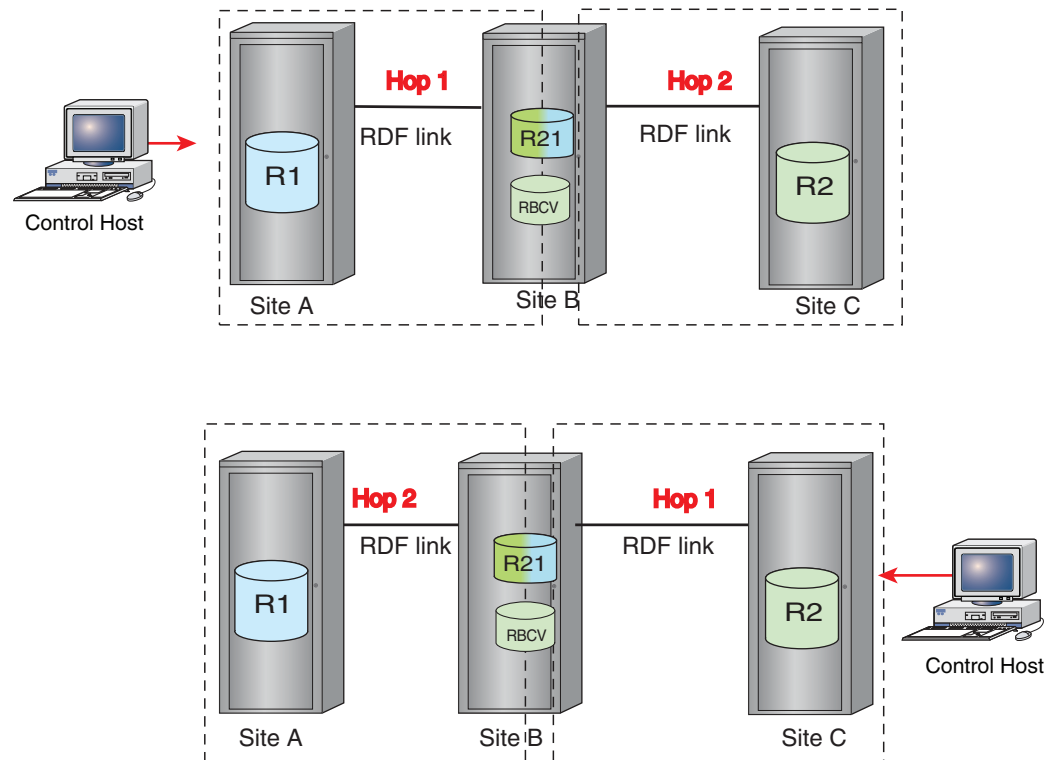
Use the `-hop2` option to control an SRDF device that is two hops away. The `-hop2` option can be used with device groups, composite groups, STDs, and local BCVs.

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 36** Location of hop-2 devices



In the image above:

- 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:`.

In the following example a composite group has 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
```

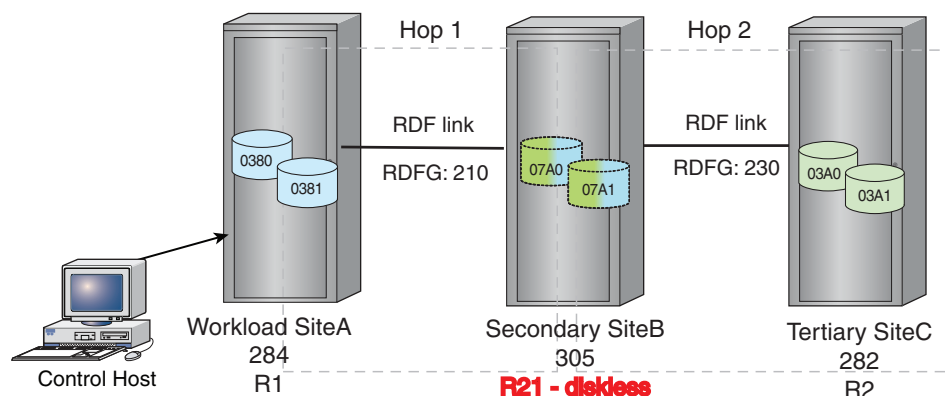
## Cascaded SRDF with EDP

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.

SRDF/EDP 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 37** 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 restrictions

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.

After these two steps, the configuration is R1 --> DLR21 --> R2.

The following table lists the SRDF modes allowed for SRDF/EDP.

**Table 27** SRDF modes allowed for SRDF/EDP

| R1 - DLR21                      | DLR21 - R2   |
|---------------------------------|--------------|
| Synchronous                     | Asynchronous |
| Adaptive copy disk <sup>a</sup> | Asynchronous |

- a. Adaptive copy mode on the first leg does not provide full time consistency of the R21 or R2 devices.

## Create cascaded SRDF/EDP pairs and set 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.

Use the command twice, once for each hop.

### Syntax

```
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.

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

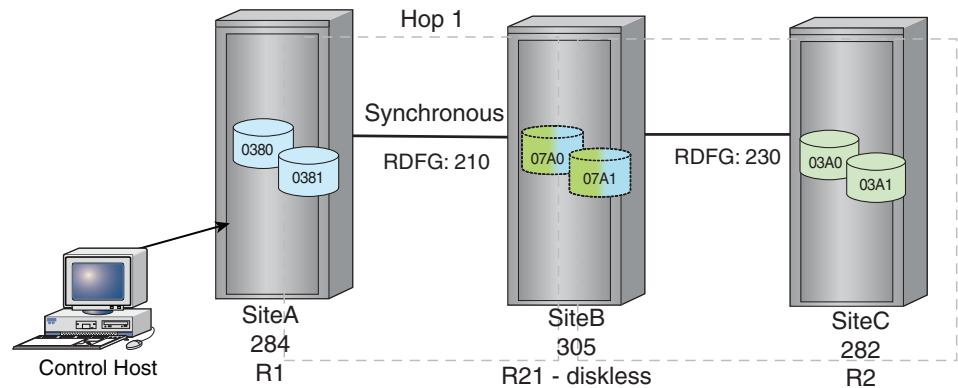
Since the R21 devices are diskless and cannot be mapped, you do not need to make the device Not Ready or Write Disabled before using the `-invalidate R2` option.

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 38** Set up first hop in cascaded SRDF with EDP

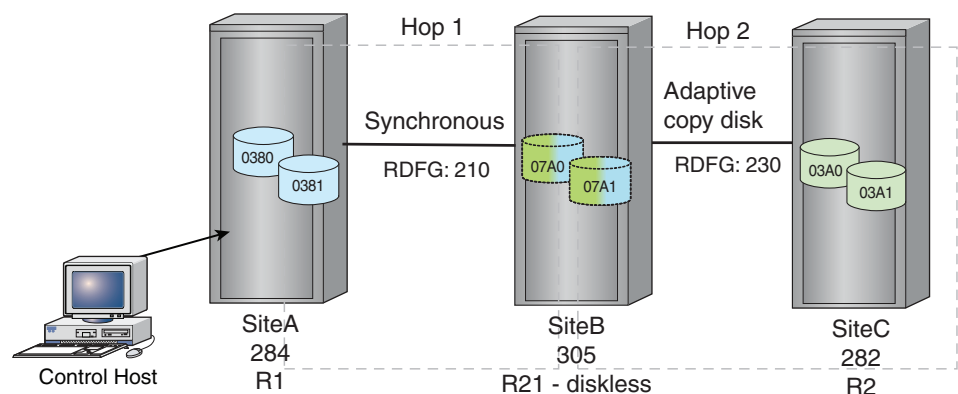


The SRDF device pairs are created and placed in synchronous mode.

- 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 39** 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
```



## 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 for diskless devices in cascaded SRDF

- 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→R2↔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 for diskless devices in cascaded SRDF

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.  
[SRDF modes in cascaded configurations on page 223](#) lists the modes allowed for cascaded SRDF configurations.  
[SRDF modes in cascaded configurations with EDP on page 224](#) 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

**NOTICE**

The adaptive copy write pending -> asynchronous combination in [SRDF modes in cascaded configurations with EDP on page 224](#) 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 for diskless devices in cascaded SRDF

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 for diskless devices in cascaded SRDF on page 233](#).
- 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 for diskless devices in cascaded SRDF

- 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.

## Create 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

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. [Add a diskless SRDF mirror on page 235](#) provides more information.

Use the `set dev` command with `symconfigure` to set attributes on diskless devices.

### Note

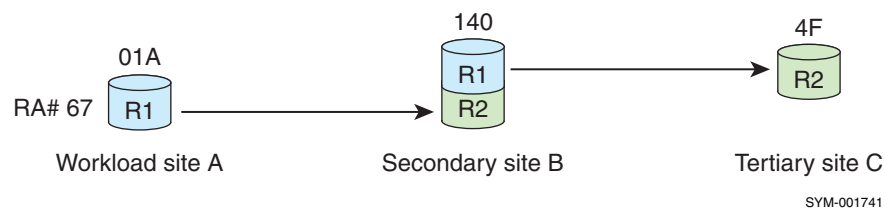
For more information about the `symconfigure` command, see the *EMC Solutions Enabler Array Management CLI User Guide*.

## Add a diskless SRDF mirror

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 40** 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.

### Syntax

Use the `symconfigure add rdf mirror` command to add both static and dynamic SRDF mirrors to diskless devices.

### 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 for diskless devices in cascaded SRDF on page 233](#).

### Examples

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
```

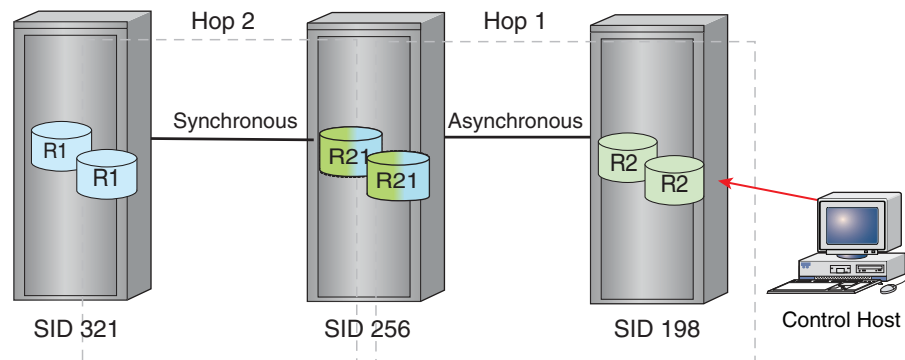
## Restart a diskless configuration

- 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 title 36:

**Figure 41** Cascaded configuration before planned failover

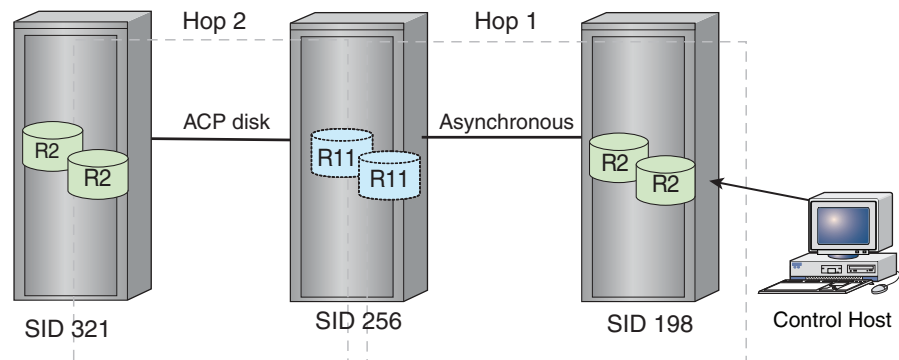


For the example session:

- 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 -gGroupName 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 GroupNamequery -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.

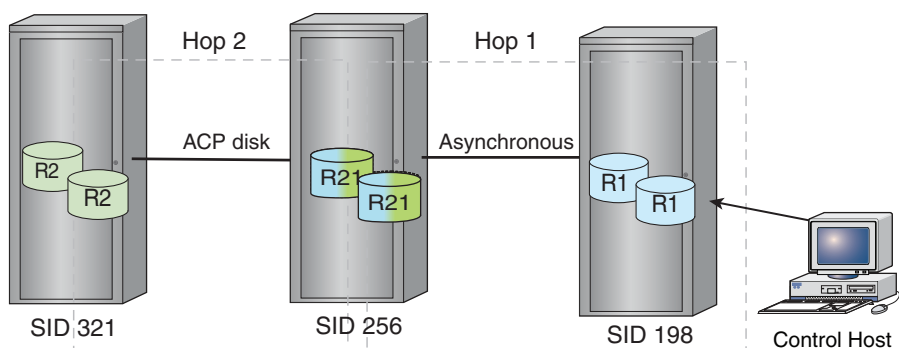
5. Use the `symrdf -g GroupName query -hop2` command to verify that the RDF Pair State for devices in the SID 321 -> SID 256 hop is Suspended.
  6. Use the `symrdf -g GroupName suspend -force` command to suspend the device pairs of the SID 256 -> SID 198 hop.
  7. Use the `symrdf -g GroupName query` 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 42** 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 43** 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.

---

## Display cascaded SRDF

You can display the following information about a cascaded SRDF configuration:

- List cascaded SRDF devices
- List diskless devices
- Query hop 2 information

### List cascaded SRDF devices

Use the `symrdf list` command with the following options to display information about cascaded SRDF devices:

#### -R21

Displays all R21 devices. This option cannot be specified in the same command with the `-R1` or `-R2` option.

#### -cascade

Lists 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.

#### -concurrent

R21 devices and the devices with which they are paired are considered concurrent devices. Use the `-concurrent` option to display these devices.

### List R21 devices

#### Syntax

Output of the `symrdf list` command includes the SRDF Mirror Type associated with the SRDF group.

#### Example

In the following example, Mirror Type is in bold text.

**`symrdf list -sid 305 -cascaded`**

Symmetrix ID: 000192600305

#### Local Device View

|       |       | R D F   S T A T E S |           |       |        |        |     |      |              |  |
|-------|-------|---------------------|-----------|-------|--------|--------|-----|------|--------------|--|
| Sym   | RDF   | STATUS              |           | MODES | 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

```

## Diskless devices

### NOTICE

`symcg`, `symsg`, or `symdev` commands used with the `relabel` option fail when the scope includes any diskless device.

### List SRDF diskless devices

#### Syntax

Use the `symrdf list` command with the `-diskless_rdf` option to view only SRDF diskless devices.

Use the `-R1`, `-R2`, `-R21`, or `-dynamic` options to display only the selected device types.

The specified diskless SRDF or SRDF capable devices are displayed.

#### Example

To display SRDF diskless devices:

```
symrdf list -diskless_rdf
```

### List all diskless devices

#### Syntax

Use the `symdev list` command with the `-dldev` option to display all configured diskless devices.

Use the `-R1`, `-R2`, `-R21`, or `-dynamic` options to display only the selected device types.

#### Example

To display all diskless devices for Symm 305:

```
symdev list -sid 305 -dldev
```

Symmetrix ID: 000192600305

| Device Name |             | Directors   | Device      |       |    |      |
|-------------|-------------|-------------|-------------|-------|----|------|
| Attribute   | Sts (MB)    |             |             |       |    | Cap  |
| 007A0       | Not Visible | ???:? ???:? | RDF21+DLDEV | Grp'd | RW | 1031 |
| 007A1       | Not Visible | ???:? ???:? | RDF21+DLDEV | Grp'd | RW | 1031 |

## Show specified diskless device

### Syntax

In the following example, output of the `symdev show` command displays the following information about the specified diskless device:

- 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

```
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
...

```

## Query hop 2 information

### Syntax

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
-----
      Standard      ST              LI      ST              C S
      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
-----
      Standard      ST              LI      ST              C S
      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(mode 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 summary

- Number of SRDF (RA) Groups — Represents the number of R1 -> R21 SRDF groups in the composite group.
- Symmetrix ID — Represents the Symmetrix ID of the R1 device.
- Hop-2 Symmetrix ID — Represents the Symmetrix ID of the R21 device.
- Hop-2 Remote Symmetrix ID — Represents the Symmetrix ID of the R2 device.
- SRDF (RA) Group Number — Represents the SRDF group of the R1 device.
- Hop-2 SRDF (RA) Group Number — Represents the 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.

## Query output detailed information

**Syntax**

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    |                    |
|------------------|-------|----|--------|--------|-------|------------------|--------|--------|-------|----------|--------------------|
| -----            |       |    |        |        |       | -----            |        |        |       | -----    |                    |
| Standard         | Sym   | T  | R1 Inv | R2 Inv | LI    | ST               |        |        |       |          |                    |
| Device           | Dev   | E  | Tracks | Tracks | K     | T                | R1 Inv | R2 Inv |       | RDF Pair |                    |
|                  |       |    |        |        | S Dev | E                | Tracks | Tracks | MDACE | STATE    |                    |
| DEV001           | 00380 | RW | 0      | 0      | 0     | RW               | 00390  | WD     | 0     | 0        | S.... Synchronized |
| DEV002           | 00381 | RW | 0      | 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    |                    |
|------------------|-------|----|--------|--------|-------|------------------|--------|--------|-------|----------|--------------------|
| -----            |       |    |        |        |       | -----            |        |        |       | -----    |                    |
| Standard         | Sym   | T  | R1 Inv | R2 Inv | LI    | ST               |        |        |       |          |                    |
| Device           | Dev   | E  | Tracks | Tracks | K     | T                | R1 Inv | R2 Inv |       | RDF Pair |                    |
|                  |       |    |        |        | S Dev | E                | Tracks | Tracks | MDACE | STATE    |                    |
| DEV001           | 00390 | RW | 0      | 0      | 0     | RW               | 003A0  | WD     | 0     | 0        | C.D.. Synchronized |
| DEV002           | 00391 | RW | 0      | 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    |                    |
|------------------|-------|----|--------|--------|-------|------------------|--------|--------|-------|----------|--------------------|
| -----            |       |    |        |        |       | -----            |        |        |       | -----    |                    |
| Standard         | Sym   | T  | R1 Inv | R2 Inv | LI    | ST               |        |        |       |          |                    |
| Device           | Dev   | E  | Tracks | Tracks | K     | T                | R1 Inv | R2 Inv |       | RDF Pair |                    |
|                  |       |    |        |        | S Dev | E                | Tracks | Tracks | MDACE | STATE    |                    |
| DEV003           | 00940 | RW | 0      | 0      | 0     | RW               | 00944  | WD     | 0     | 0        | S.... Synchronized |
| DEV004           | 00941 | RW | 0      | 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    |                    |
|------------------|-------|----|--------|--------|-------|------------------|--------|--------|-------|----------|--------------------|
| -----            |       |    |        |        |       | -----            |        |        |       | -----    |                    |
| Standard         | Sym   | T  | R1 Inv | R2 Inv | LI    | ST               |        |        |       |          |                    |
| Device           | Dev   | E  | Tracks | Tracks | K     | T                | R1 Inv | R2 Inv |       | RDF Pair |                    |
|                  |       |    |        |        | S Dev | E                | Tracks | Tracks | MDACE | STATE    |                    |
| DEV003           | 00944 | RW | 0      | 0      | 0     | RW               | 00942  | WD     | 0     | 0        | C.D.. Synchronized |
| DEV004           | 00945 | RW | 0      | 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 — Represents the 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 — Represents the 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 — Represents the 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

Each R21->R2 SRDF group is reported separately.

# CHAPTER 9

## SRDF/Star Operations

This chapter describes the following topics.

|                                                   |     |
|---------------------------------------------------|-----|
| • SRDF/Star operations overview .....             | 246 |
| • SRDF/Star states and operations .....           | 251 |
| • SRDF/Star operations summary .....              | 263 |
| • Configure and bring up SRDF/Star .....          | 267 |
| • Basic SRDF/Star operations .....                | 283 |
| • SRDF/Star consistency group operations .....    | 286 |
| • Recovery operations: Concurrent SRDF/Star ..... | 297 |
| • Workload switching: Concurrent SRDF/Star .....  | 301 |
| • Recovery operations: Cascaded SRDF/Star .....   | 318 |
| • Workload switching: Cascaded SRDF/Star .....    | 322 |
| • Reconfiguration operations .....                | 333 |
| • SRDF/Star configuration with R22 devices .....  | 345 |

## SRDF/Star operations overview

SRDF/Star deployments include three geographically dispersed data centers in a triangular topology. 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 Engenuity 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:

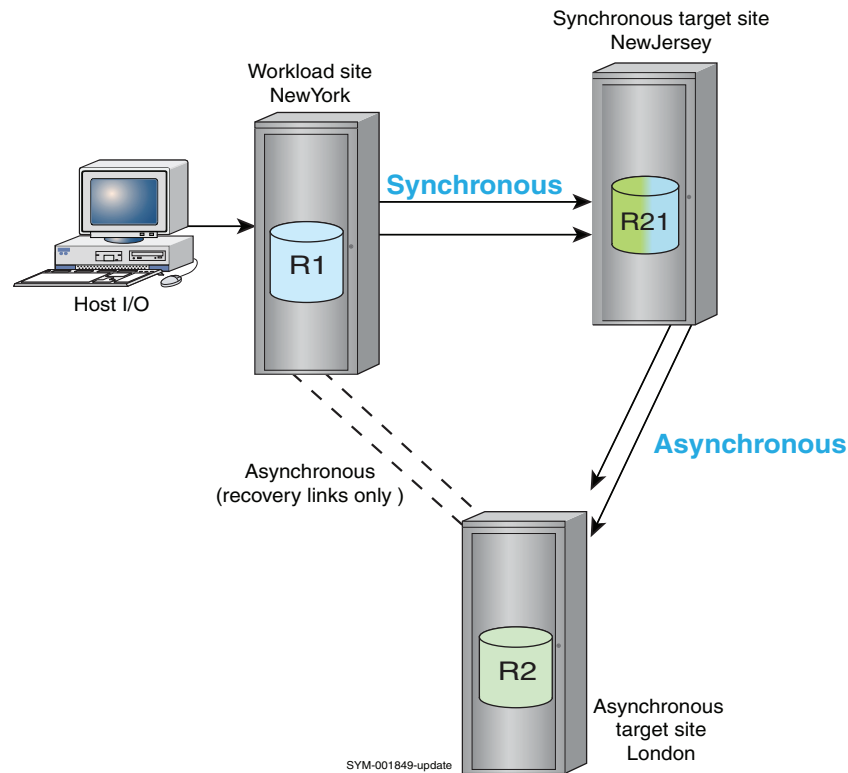
- Cascaded SRDF/Star
- Concurrent SRDF/Star
- Concurrent SRDF/Star with R22 devices

## 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 44** Cascaded SRDF/Star configuration

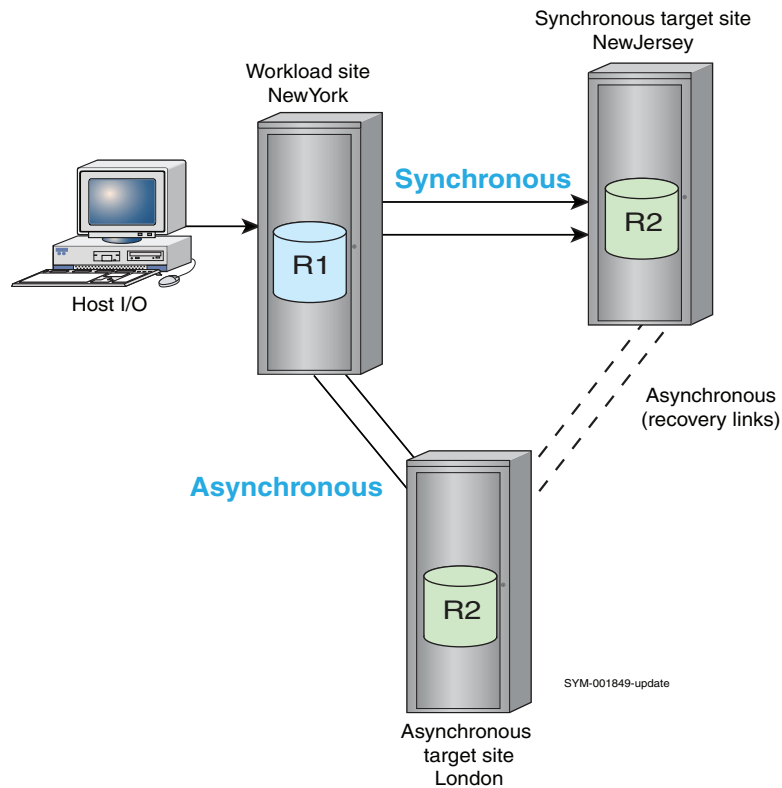
In cascaded SRDF/Star, the synchronous target site is always more current than the asynchronous target site, but it is possible to determine which site's data to use for recovery.

## 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 45** Concurrent SRDF/Star configuration

Data transfer from the workload site is:

- Synchronous to the nearby target site (New York) 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, it is possible to determine which target site contains the most current data.

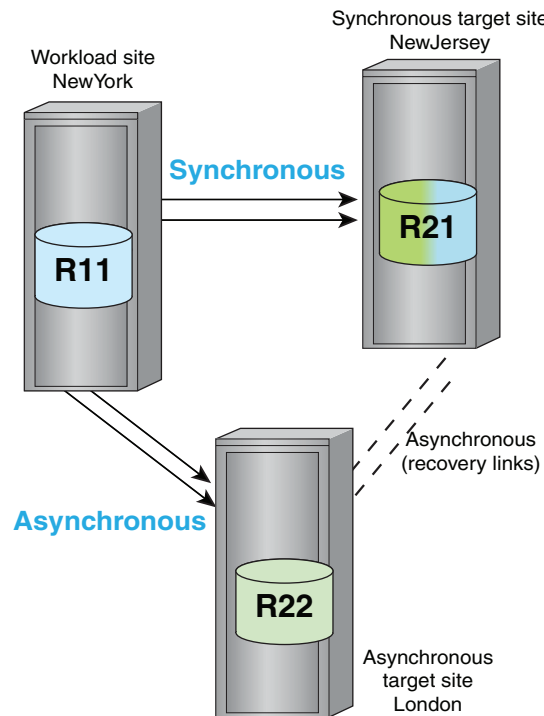
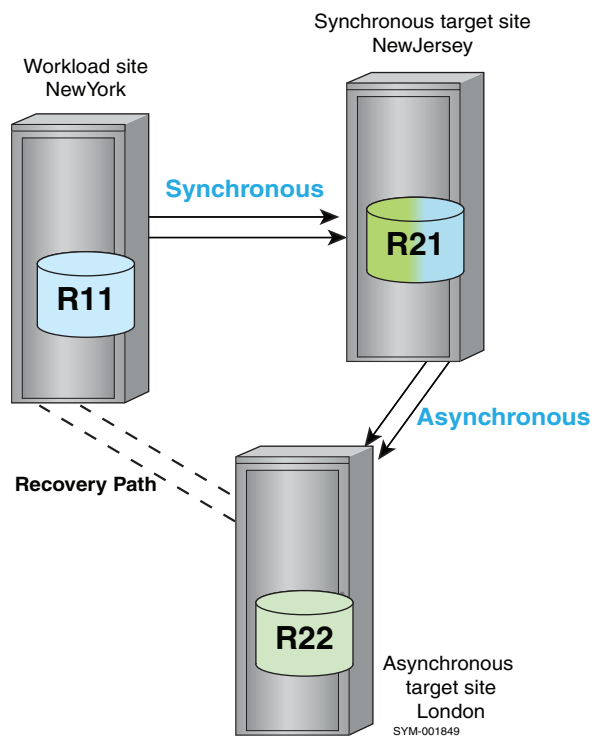
## Concurrent SRDF/Star with R22 devices

### 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 46** Typical concurrent SRDF/Star with R22 devices**Figure 47** Typical cascaded SRDF/Star with R22 devices

R11 and 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, it is possible to determine which of the target sites holds the more current data and switch workload operations to that site.
- Devices can be added to an SRDF consistency group or removed from an SRDF consistency group to maintain data consistency without interrupting the workload.

## SRDF/Star restrictions

- Devices that are part of an SRDF/Metro configuration cannot at the same time be part of an SRDF/Star configuration.
- 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.

## SRDF/Star states and operations

The state of the SRDF/Star environment determines possible operations and includes the following:

- The SRDF/Star state of the 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 28** 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.<br><br>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.                                                                                                                                                                      |

**NOTICE**

The configuration must be in the Star Protected state in order to have SRDF/Star 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 29** SRDF/Star target site states

| State        | Description                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Disconnected | May indicate that there is no data flow between the workload site and the target sites.<br><br><b>Note</b><br>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. |

**Table 29** SRDF/Star target site states (continued)

| State     | Description                                                                                                                                                                                                                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connected | There is data flow between the sites.<br>The target site is not necessarily synchronized with the workload site.                                                                                                                                                                                                                     |
| Protected | There is data flow between the sites.<br>Dependent write consistency of the data at the target site is assured.                                                                                                                                                                                                                      |
| Halted    | There is no data flow between the sites.<br>There is no data protection at the target site relative to the workload site.<br>The data at each site is the same.                                                                                                                                                                      |
| Isolated  | There is no data flow between the sites.<br>The devices at the target site are read/write enabled to their local host.                                                                                                                                                                                                               |
| PathFail  | <p>There is no data flow between the sites.</p> <hr/> <p><b>Note</b></p> <p>Occurs only if the specified target was in a Protected state.</p> <hr/> <p>The PathFail;CleanReq state indicates that the <code>cleanup</code> operation is required to perform MSC cleanup on the asynchronous target before it will be consistent.</p> |

## SRDF/Star site configuration transitions

In the following discussion, the initial configuration 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.

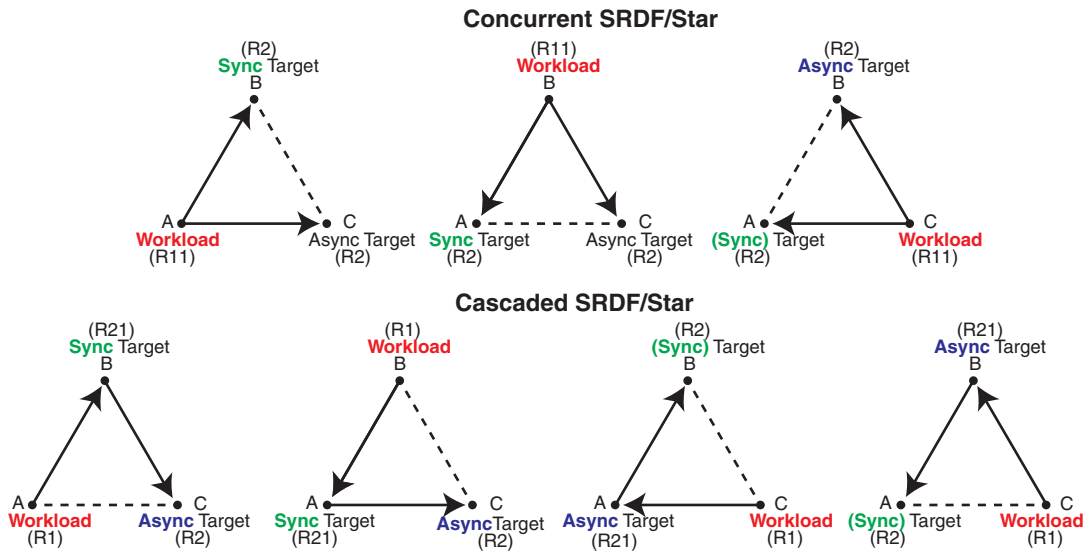
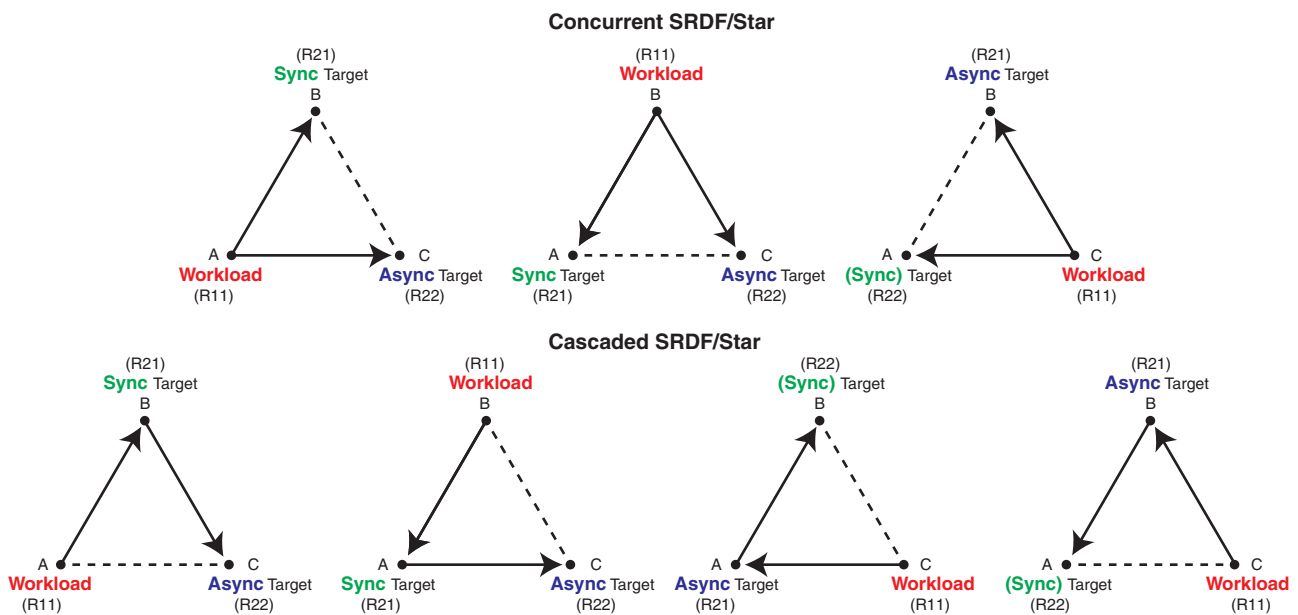
### NOTICE

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.

**Note**

In the following diagrams, one of the targets is labeled as the (Sync) target in order to differentiate between the two target sites.

**Transitions without concurrent devices****Figure 48** Site configuration transitions without concurrent devices**Transitions with concurrent devices****Figure 49** Site configuration transitions with concurrent devices**SRDF/Star operation categories**

SRDF/Star operations can be broken into four categories.

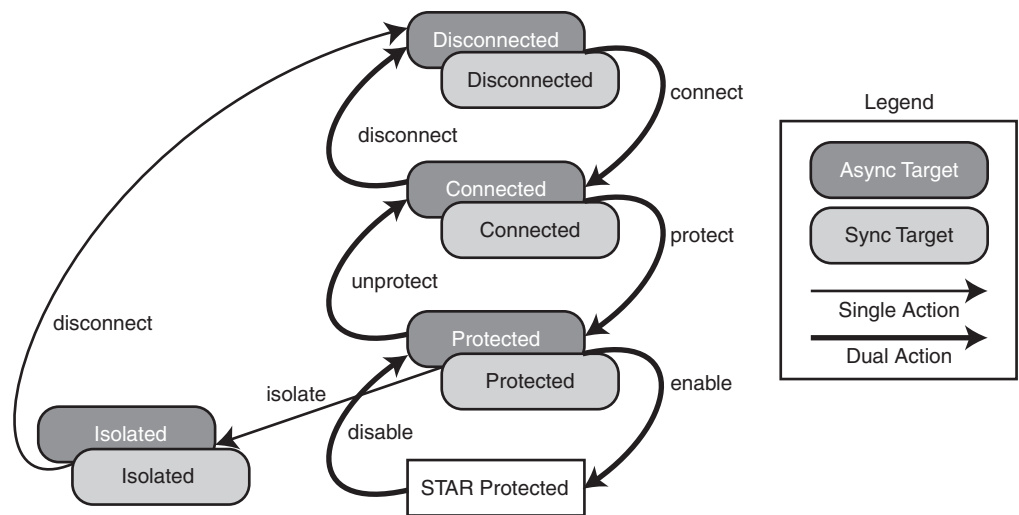
**Table 30** SRDF/Star operation categories

| Operation Category         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Normal operations          | Used to configure and setup SRDF/Star to achieve SRDF/Star protection.<br>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.<br>Transient faults do not disrupt production at the workload site, so these operations can be executed at the workload site.                                                                                                                                                                                                                                                                                                                                                          |
| Switch operations          | <ul style="list-style-type: none"> <li>Planned:<br/>Used to move the production workload to a new site with a planned procedure.<br/><br/>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.</li> <li>Unplanned:<br/>Used to recover from faults caused by the loss of a workload site.<br/><br/>The loss of a workload site requires an unplanned switch of the workload to one of the target sites.</li> </ul> |
| Reconfigure operations     | <ul style="list-style-type: none"> <li>Planned:<br/>Transitions the SRDF/Star setup from concurrent SRDF to cascaded SRDF or vice versa as part of a planned event.</li> <li>Unplanned:<br/>Transitions the SRDF/Star setup from concurrent SRDF to cascaded SRDF or vice versa after a failure.</li> </ul> <p>Reconfigure operations can be used to resolve a transient fault or as part of a switch operation.</p>                                                                                                                                                                          |

## Required states for operations: Concurrent SRDF/Star

### Normal operations

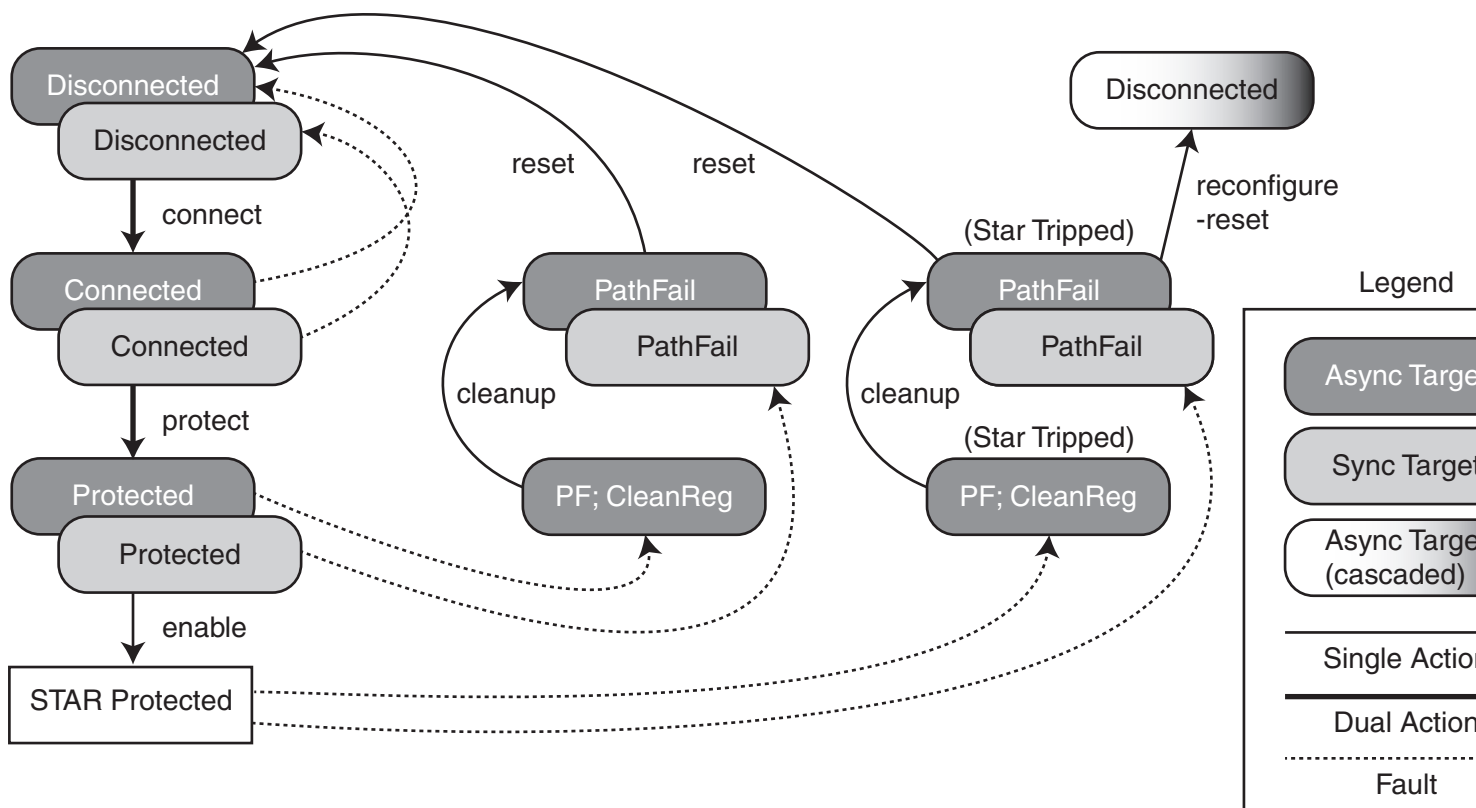
The following image shows the normal operations that are available from each state.

**Figure 50** 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

The following image shows the transient fault operations that are available from each state.

**Figure 51** 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

If the workload site fails, an unplanned **switch** operation is required to move the production workload to one of the target sites.

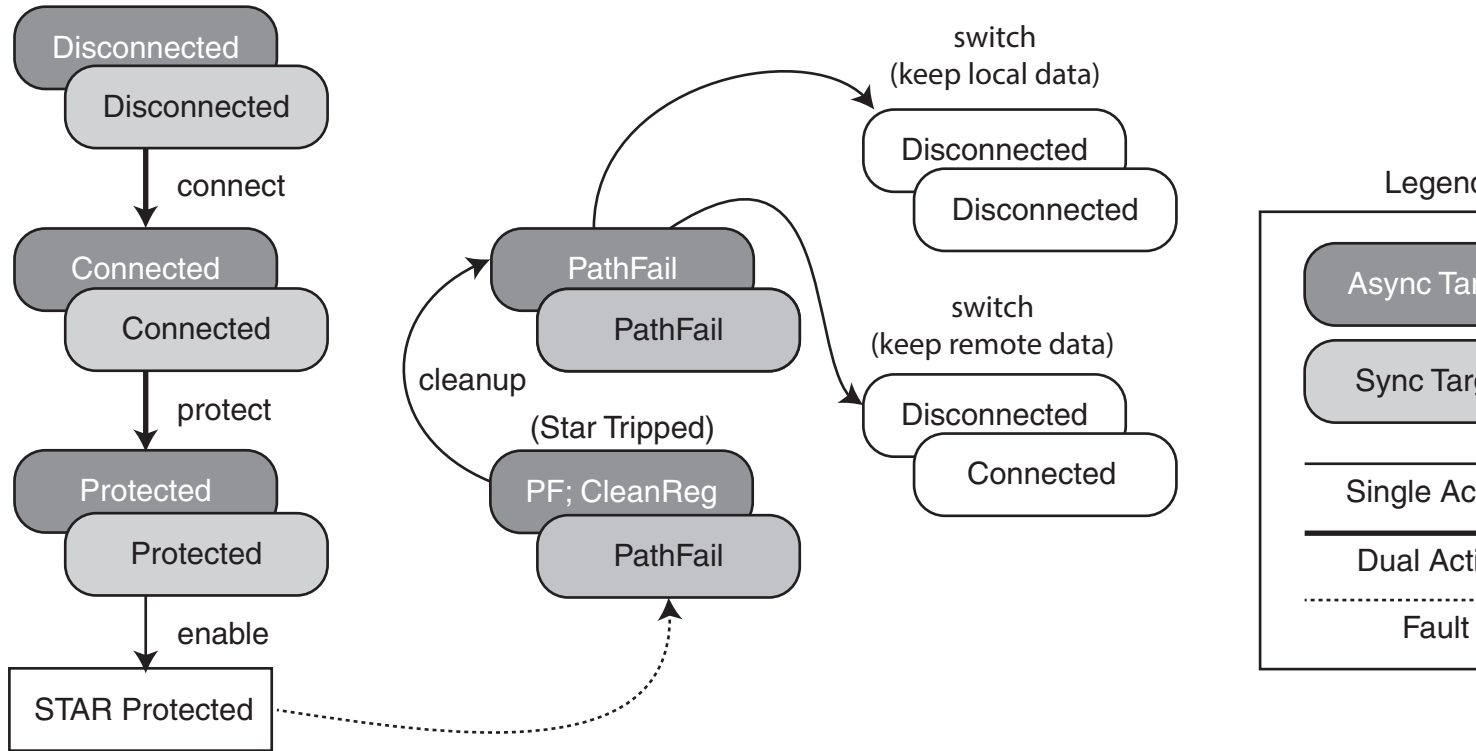
The following image shows the unplanned switch operations that are available from each state.



**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** Concurrent SRDF/Star: unplanned switch operations



When switching to a target site, the options are as follows:

- 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

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.

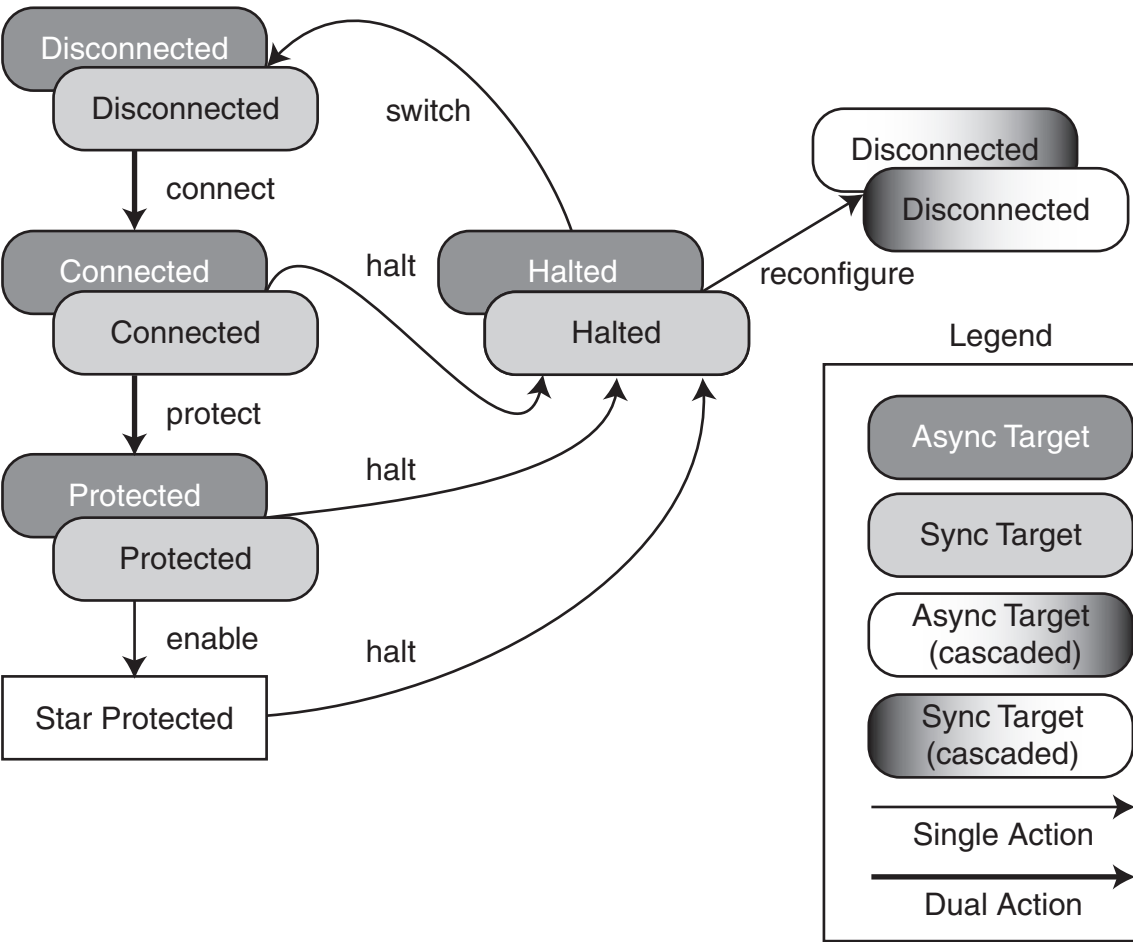
The `halt` operation write-disables the R1 devices, drains the data to the two target sites, and makes the data at all three sites the same.

### NOTICE

Before initiating the `halt` operation, 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 following image shows the planned switch operations that are available from each state.

**Figure 53** Concurrent SRDF/Star: planned switch operations



## Required states for operations: Cascaded SRDF/Star

### Normal operations

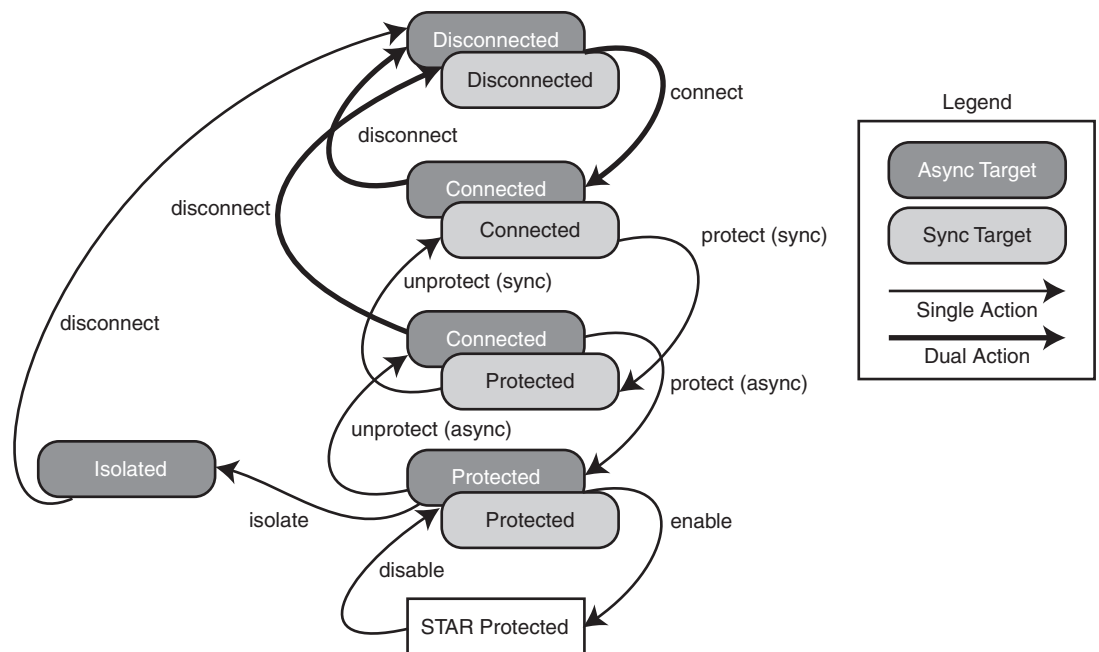
In Cascaded SRDF/Star, the consistency of the asynchronous site data is dependent on the consistency of the synchronous site data.

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

The synchronous target site can be isolated if the asynchronous target site has a target site state of Disconnected, Isolated, or PathFail.

The following image shows the normal operations that are available from each state.

**Figure 54** 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 continue production at the workload site without remote replication. When the outage is restored, you can then reset the synchronous target.

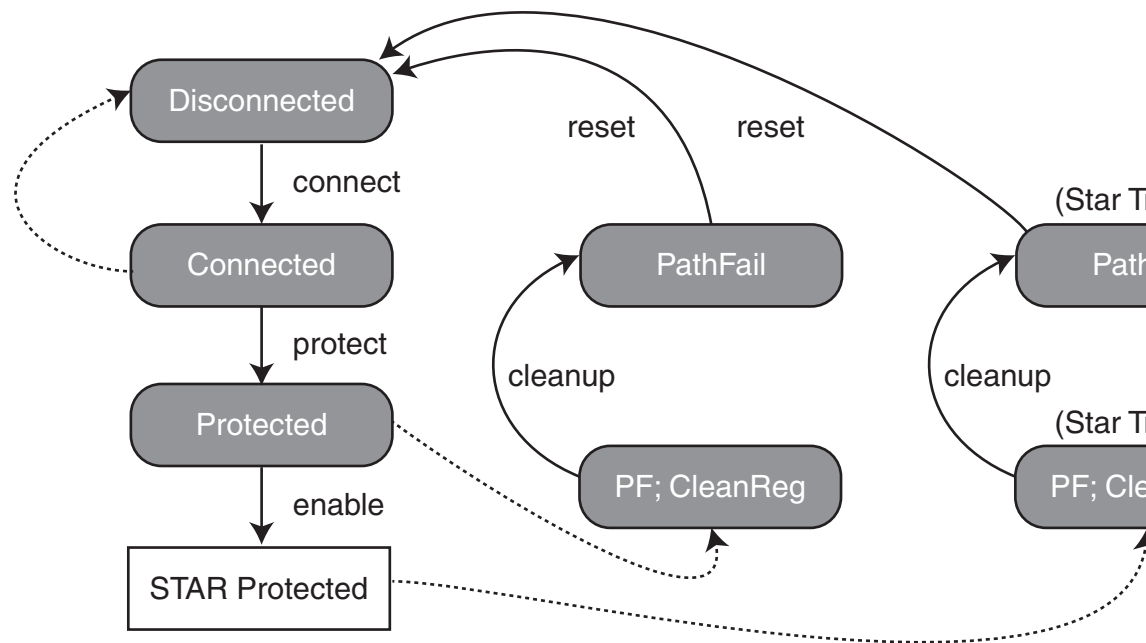
The following image shows the transient fault operations that are available from each state after the loss of the asynchronous target site.

---

### Note

This diagram assumes that the synchronous target stayed protected during the fault.

---

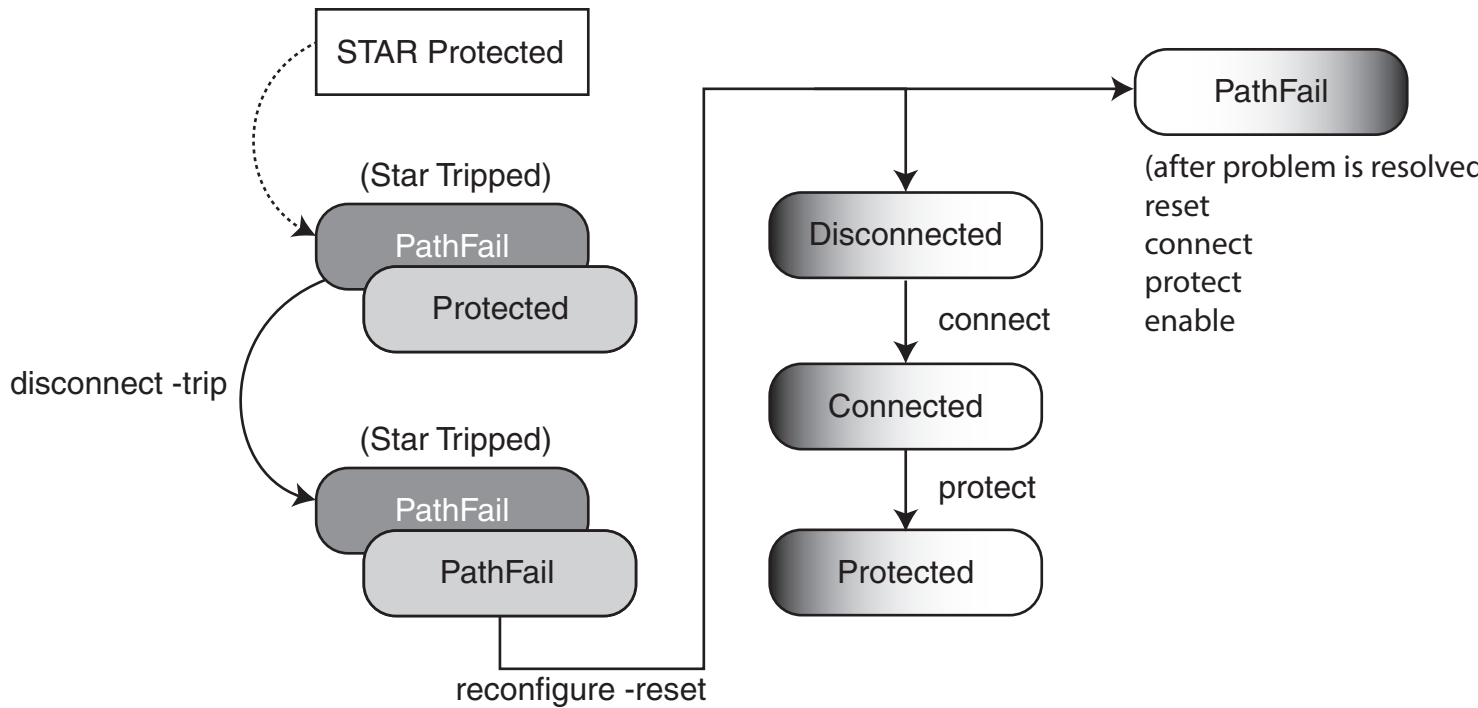
**Figure 55** Cascaded SRDF/Star: transient fault operations (asynchronous loss)

- 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 SRDF/Star to Concurrent SRDF/Star

Reconfigure Cascaded SRDF/Star to Concurrent SRDF/Star to have remote replication immediately after the synchronous target is lost.

The following image 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 56** Cascaded SRDF/Star: transient fault operations (synchronous loss)

## Unplanned switch operations

In Cascaded/SRDF, if the workload site fails, an unplanned `switch` operation is required to move the production workload to one of the target sites.

- To switch production to the synchronous target site, convert the configuration to Concurrent SRDF/Star. Only local data can be kept because the local data is ahead of the data at the asynchronous target site.
- When switching production to the asynchronous target site, the local data or the data at the synchronous target site can be kept.

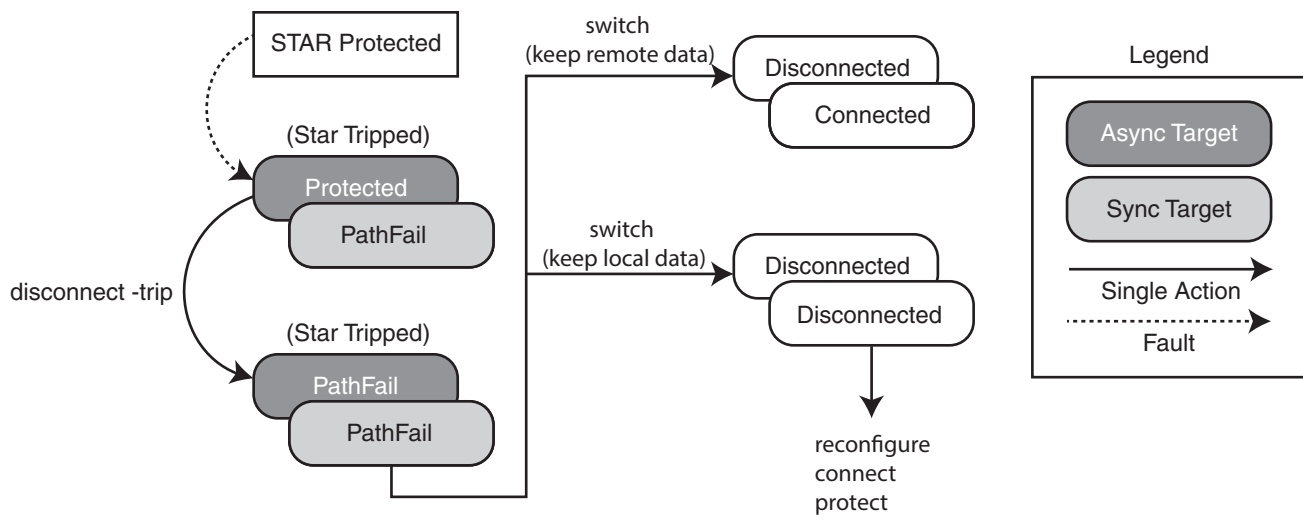
The following image shows unplanned switch operations that are available from each state.

---

### 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 57** Cascaded SRDF/Star: unplanned switch operations

## SRDF/Star operations summary

**Table 31** SRDF/Star control operations

| Control operation                                                                                                                                                        | symstar argument                                                                                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                  | Workload or target |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <a href="#">Configure and bring up SRDF/Star on page 267</a>                                                                                                             | symrdf list<br>symcg create<br>symcg addall<br>symcg set<br>symstar setup<br>symstar buildcg<br>symstar connect<br>symstar protect<br>symstar enable | Sample procedure showing the basic steps to configure and activate the SRDF/Star environment.                                                                                                                                                                                                                                                                                |                    |
| <a href="#">Displaying the symstar configuration on page 280</a><br><a href="#">symstar show command on page 281</a><br><a href="#">symstar list command on page 283</a> | query<br>show<br>list                                                                                                                                | <ul style="list-style-type: none"> <li>Displays the status of a given SRDF/Star site configuration.</li> <li>Displays the contents of the internal definition for a given SRDF/Star site configuration.</li> <li>Lists each SRDF/Star composite group configuration, including workload name, mode of operation, CG and Star states, and target names and states.</li> </ul> | W/T                |
| <a href="#">Isolate SRDF/Star sites on page 284</a>                                                                                                                      | isolate                                                                                                                                              | Isolates one target site from the SRDF/Star configuration and makes its R2 devices read/write enabled to their hosts.                                                                                                                                                                                                                                                        | W                  |
| <a href="#">Unprotect target sites on page 285</a>                                                                                                                       | unprotect                                                                                                                                            | Disables SRDF/Star consistency protection to the specified target site.                                                                                                                                                                                                                                                                                                      | W                  |
| <a href="#">Halt target sites on page 285</a>                                                                                                                            | 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                |
| <a href="#">Clean up metadata on page 285</a>                                                                                                                            | cleanup                                                                                                                                              | Cleans up internal meta information and cache at the remote site after a failure at the workload site.                                                                                                                                                                                                                                                                       | T                  |
| <a href="#">SRDF/Star consistency group operations on page 286</a>                                                                                                       | modifycg                                                                                                                                             | Maintains consistency protection when adding or removing device pairs from an SRDF/Star consistency group.                                                                                                                                                                                                                                                                   | W                  |
| <a href="#">Upgrade an existing SRDF/Star environment</a><br><a href="#">Transition SRDF/Star to use R22 devices on page 346</a>                                         | 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                  |
| <a href="#">Begin SRDF synchronization</a>                                                                                                                               | connect                                                                                                                                              | Starts the SRDF data flow in adaptive copy disk mode.                                                                                                                                                                                                                                                                                                                        | W                  |

**Table 31** SRDF/Star control operations (continued)

| Control operation                                                                                                                                                                                                                                                                                                                    | symstar argument | Description                                                                                                                                                             | Workload or target |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Enable full SRDF/Star protection                                                                                                                                                                                                                                                                                                     | enable           | Enables complete SRDF/Star consistency protection across the three sites.                                                                                               | W                  |
| <a href="#">SRDF/Star consistency group operations on page 286</a>                                                                                                                                                                                                                                                                   | protect          | Synchronizes devices between the workload and target sites and enables SRDF/Star consistency protection to the specified target site.                                   | W                  |
| Change the SRDF/Star replication path<br><a href="#">Reconfiguring mode: cascaded to concurrent on page 334</a> , <a href="#">Reconfiguring cascaded paths on page 338</a> , <a href="#">Reconfiguring mode: concurrent to cascaded on page 340</a> , <a href="#">Reconfigure mode without halting the workload site on page 344</a> | 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                  |
| Reset after a transient failure<br><a href="#">Recovery operations: Concurrent SRDF/Star on page 297</a> , <a href="#">Recovery operations: Cascaded SRDF/Star on page 318</a>                                                                                                                                                       | 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<br><a href="#">Workload switching: Concurrent SRDF/Star on page 301</a> , <a href="#">Unplanned workload switching: cascaded SRDF/Star on page 324</a> <a href="#">Unplanned workload switching to asynchronous target site: Cascaded SRDF/Star on page 330</a>                          | switch           | Transitions workload operations to a target site after a workload site failure or as part of a planned event.                                                           | T                  |
| Verify that the given site or SRDF/Star setup is in the desired state<br><a href="#">Displaying the symstar configuration on page 280</a>                                                                                                                                                                                            | 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 32** symstar command options

| 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_r2l_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_r2l_rdfg option.                                                                                                                                                                                                                           |
| -cleanreq      | Verifies the site is in the PathFail state and needs cleaning.                                                                                                                                                                                                                                                                                                                                                                                                |
| -connected     | Verifies the site is in the <code>connected</code> state.                                                                                                                                                                                                                                                                                                                                                                                                     |
| -devs          | Specifies the ranges of devices to add or remove.                                                                                                                                                                                                                                                                                                                                                                                                             |
| -disconnected  | Verifies the site is in the <code>disconnected</code> state.                                                                                                                                                                                                                                                                                                                                                                                                  |
| -distribute    | Performs an automatic SRDF/Star definition file distribution. This form of setup does not disrupt an active protected SRDF/Star setup.                                                                                                                                                                                                                                                                                                                        |
| -full          | <ul style="list-style-type: none"> <li>Used by <code>reconfigure</code>, <code>switch</code>, and <code>connect</code>. Performs a full SRDF resynchronization if SRDF incremental resynchronization is not available.</li> <li>Used by the <code>list</code> action to display full names instead of abbreviations.</li> </ul>                                                                                                                               |
| -halted        | Verifies the site is in the <code>halted</code> state.                                                                                                                                                                                                                                                                                                                                                                                                        |
| -haltfail      | Verifies the site is in the <code>haltfail</code> state.                                                                                                                                                                                                                                                                                                                                                                                                      |
| -haltstarted   | Verifies the site is in the <code>haltstarted</code> 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 <code>isolated</code> state.                                                                                                                                                                                                                                                                                                                                                                                                      |
| -keep_data     | Identifies which site's data is retained when used with the <code>switch</code> and <code>connect</code> 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 <code>connect</code> 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 <code>list</code> action.                                                                                                                                                                                                                                                                                                                                                                          |
| -offline       | Obtains the data strictly from the configuration database. No connections are made to any arrays. The <code>symstar</code> 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).                                                                                                                                                                                                                                                                                                                                                                                                     |

Table 32 symstar command options (continued)

| Command option   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -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 <code>pathfail</code> state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| -pathfailinprog  | Verifies the site is in the <code>pathfailinprog</code> state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| -protected       | Verifies the site is in the <code>protected</code> state. If -site is not specified, verifies that SRDF/Star is in the <code>protected</code> state.                                                                                                                                                                                                                                                                                                                                                                                       |
| -noprompt        | Suppresses the message asking you to confirm an SRDF control operation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| -reload_options  | <p>Reads the specified options file to update the SRDF/Star definition file when using the setup action.</p> <hr/> <p><b>Note</b></p> <p>Do not change any <code>SITE_NAME</code> values with this option.</p> <hr/>                                                                                                                                                                                                                                                                                                                       |
| -remote          | <p>Indicates the remote data copy flag. Used with the <code>connect</code> action when keeping remote data and the concurrent link is ready. Data is also copied to the concurrent SRDF mirror.</p> <hr/> <p><b>Note</b></p> <p>Not required if the concurrent link is suspended.</p> <hr/>                                                                                                                                                                                                                                                |
| -remove          | <ul style="list-style-type: none"> <li>For the <code>reconfigure</code> action, specifies the sites on which the SRDF pairs are removed.</li> <li>For the <code>setup</code> action, specifies that all SRDF/Star mode settings for all SRDF groups be set to off if the CG is defined in the <code>symapi</code> database, and to remove all SRDF/Star metadata associated with the group.</li> <li>For the <code>modifycg</code> action, indicated to remove the specified devices from the SRDF/Star CG to the staging area.</li> </ul> |
| -reset           | <p>Performs a reset action on the path when the <code>reconfigure</code> action is issued.</p> <ul style="list-style-type: none"> <li>When used with the <code>halt</code> action, allows the application to be restarted at the same site after the halt command has completed or failed.</li> <li>When used with the <code>configure</code> action, specifies the element of the reset operation.</li> </ul>                                                                                                                             |
| -site            | Specifies the SiteName to apply the given action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| -stg_r21_rdfg    | For <code>modifycg</code> 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 <code>-cg_r21_rdfg</code> option.                                                                                                                                                                                                                             |
| -stg_rdfg        | For the <code>modifycg</code> 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 <code>-cg_rdfg</code> option. This association is based on their order in this option and <code>-cg_rdfg</code> .                                                                                                                                                                                 |
| -trip            | Transitions the site to <code>pathfail</code> state when used with disconnect action.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| -tripped         | Verifies SRDF/Star is in the <code>tripped</code> state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| -trip_inprogress | Verifies SRDF/Star is in the <code>trip_inprogress</code> state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

**Table 32** symstar command options (continued)

| Command option | Description                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -unprotected   | Verifies the site is in the <code>unprotected</code> state. If <code>-site</code> is not specified, verifies SRDF/Star is in the <code>unprotected</code> 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 <code>symstar</code> 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

- The `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.
- The `symstar isolate` command cannot be issued to the synchronous target site since it is not possible to isolate only the synchronous target site.

## 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:

### Procedure

1. Verify the SRDF/Star host is locally connected to *only one* of the three sites.  
[Step 1: Verify SRDF/Star host connectivity on page 268](#)
2. Verify the settings for each array to be included in the SRDF/Star configuration.  
[Step 2: Verify array settings on page 269](#)
3. Create a composite group at the workload site.  
[Step 3: Create an SRDF/Star composite group on page 269](#)
4. Create an SRDF/Star options file containing specific parameters for the setup procedure.

[Step 4: Create the SRDF/Star options file on page 274](#)

5. Issue 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 276](#)

6. Optionally, issue the `symstar buildcpg` command to build matching R2 or R21 composite groups at the target sites.

[Step 6: Create R2 or R21 composite groups on page 277](#)

7. Optionally, add BCVs to the SRDF/Star configuration.

[Step 7: \(Optional\) Add BCV devices to the SRDF/Star configuration on page 278](#)

8. Bring up the SRDF/Star configuration.

[Step 8: Bring up the SRDF/Star configuration on page 279](#)

To perform SRDF/Star operations with access control enabled, the SRDF, BASECTRL, BASE, and BCV access types are required.

*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.

**Procedure**

- Issue 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 |               |          |         |           |          |          |
|-------------------|---------------|----------|---------|-----------|----------|----------|
| SymmID            | Attachment    | Model    | Mcode   | Cache     | Num Phys | Num Symm |
|                   |               |          | Version | Size (MB) | Devices  | Devices  |
| 000194901217      | <b>Local</b>  | VMAX-1SE | 5876    | 28672     | 369      | 6689     |
| 000194901235      | <b>Remote</b> | VMAX-1SE | 5876    | 28672     | 0        | 6890     |

|              |        |          |      |       |   |      |
|--------------|--------|----------|------|-------|---|------|
| 000194901241 | Remote | VMAX-1SE | 5876 | 28672 | 0 | 7007 |
|--------------|--------|----------|------|-------|---|------|

## Step 2: Verify array settings

### Procedure

- Verify that each array within SRDF/Star uses dynamic SRDF devices.  
Issue 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).  
Issue the `symcfg list -sid SID -rdfg all` command to display SRDF group-level settings for a specific group or all groups including director configuration.
- Issue 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
- Issue 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:

### Procedure

1. Create an RDF1 type composite group on the control host for the array at the workload site (NewYork).
2. Enable consistency protection for the composite group.

This step varies depending on the topology of the SRDF configuration:

- For Concurrent SRDF/Star, proceed to [Step 3, option A: Create a composite group in Concurrent SRDF/Star on page 270](#).
- For Cascaded SRDF/Star, skip to [Step 3, option B: Create a composite group in Cascaded SRDF/Star on page 272](#).

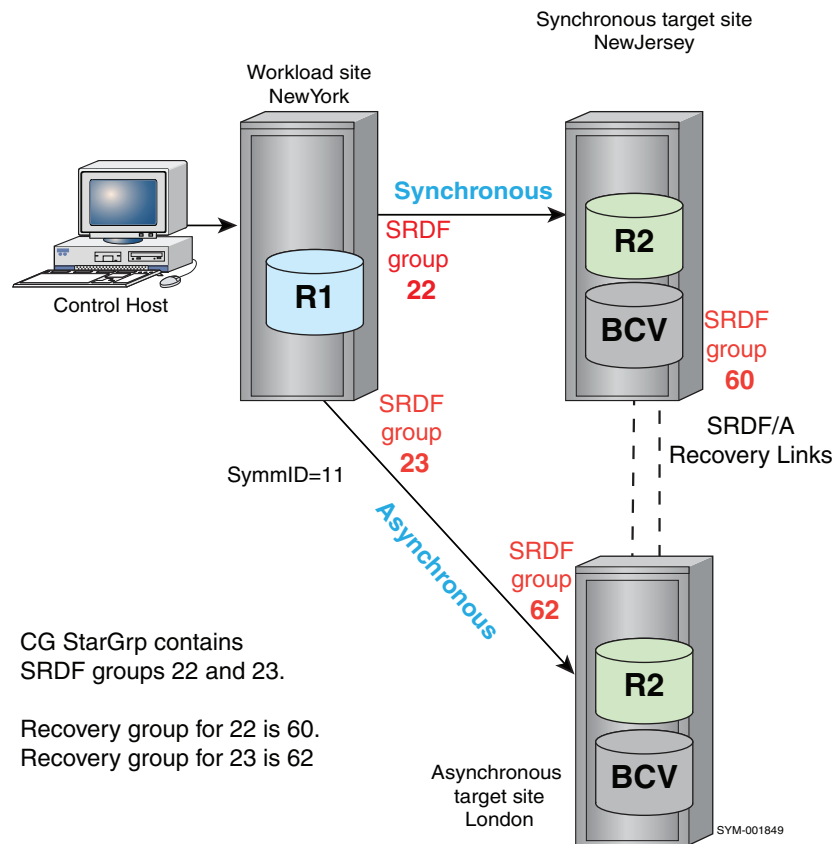
### Step 3, option A: Create a composite group in Concurrent SRDF/Star

Follow these steps if the 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 58** 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 `symcgr -cg` command.

Complete 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:

#### Procedure

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**

Specify 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 11 and 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 274 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 274](#).

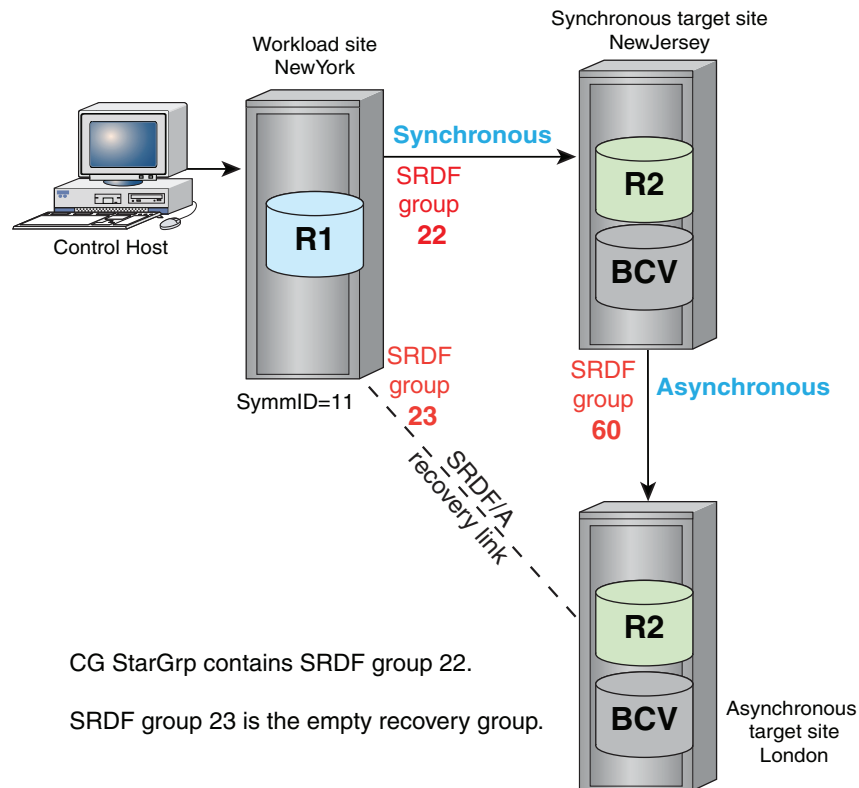
### Step 3, option B: Create a composite group in Cascaded SRDF/Star

Follow these steps if the 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`
- The asynchronous target site is `London`



**Figure 59** Cascaded SRDF/Star setup using the StarGrp composite group

Complete 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 cascaded environment:

#### Procedure

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

Specify 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

Specify 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.

---

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 274](#) provides more information.

5. For each source SRDF group 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 59 on page 273](#), the recovery SRDF group is the empty SRDF group 23 configured between the `NewYork` synchronous 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

### Description

The SRDF/Star options file specifies the names of each SRDF/Star site and other required parameters.

### Syntax

The SRDF/Star options file must conform to the following syntax:

```
SYMCL_STAR_OPTION=Value
```

You can add comment lines that begin with "#".

```
#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_ASYNC_TARGET_RDF_MODE=ACP|ASYNC

```

---

### Note

If the options file contains the `SYMCLI_STAR_COMPATIBILITY_MODE` parameter, it must be set to `v70`.

---

## Options

### 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 SRDF/Star configuration.

---

### Note

There are no SRDF group names for the asynchronous site in a cascaded configuration.

---

### NumberTracks

Maximum number of invalid tracks allowed for SRDF/Star to transition from adaptive copy mode to synchronous or asynchronous mode. SRDF/Star will wait until the number of invalid tracks is at or below the `NumberTracks` value 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 timeout 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 SDDF (Symmetrix Differential Data Facility) sessions on both the synchronous and asynchronous 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 is the 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\_ASYNC\_TARGET\_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.
- SYNC - The SRDF mode between the workload site and asynchronous target site remains asynchronous at the end of the `symstar unprotect` action.

**Examples**

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_ASYNC_TARGET_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****Description**

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 the SRDF/Star options file, and then automatically written in an internal format to the SFS on a array at each site.

**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]
```

## Options

### -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.

Specify 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.

### 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

Specify the `-distribute` option from the workload site when both target sites are reachable.

---

## Examples

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

### Description

Once the setup is complete and the SRDF/Star definition file is distributed to the SFS at the other sites, issue the `symstar buildcg` command to create the matching R2 or R21 composite groups needed for recovery operations at the synchronous and asynchronous target sites.

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, manually add the BCVs to the synchronous and asynchronous composite groups.

The next step varies depending on whether BCV devices are used:

- If BCV devices are used to retain a consistent restartable image of the data, proceed to [Step 7: \(Optional\) Add BCV devices to the SRDF/Star configuration on page 278](#).
- If not, skip to [Step 8: Bring up the SRDF/Star configuration on page 279](#).

### Syntax

```
symstar -cg CgName [-noprompt]
          buildcg -site SiteName [-update]
```

### Examples

To create the matching composite groups for `NewJersey` and `London`:

```
symstar -cg StarGrp buildcg -site NewJersey
symstar -cg StarGrp buildcg -site London
```

### Restrictions

- 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, manually add the BCVs to the synchronous and asynchronous composite groups.

## Step 7: (Optional) Add BCV devices to the SRDF/Star configuration

### Description

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 281](#) 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.

If the system state is:

- **Connected** - The devices are already read/write (RW) on the SRDF link. Skip to Step 3.
- **Disconnected** - Issue 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
```

#### Options

##### **connect**

Sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links, but does not wait for synchronization.

##### **protect**

Transitions to the correct SRDF mode (synchronous or asynchronous), enables SRDF consistency protection, waits for synchronization, and sets the STAR mode indicators.

#### **enable**

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 the symstar configuration

This section describes output of the following commands:

- `symstar query`
- `symstar show`
- `symstar list`

See also

- [Commands to display, query, and verify SRDF configurations on page 36](#)
- `symrdf list` command options

### symstar query command

#### **Description**

The `symstar query` command displays the local and remote array information and the status of the SRDF pairs in the composite group.

---

#### **Note**

Using the `-detail` option with `symstar query` includes extended information, such as the full Symmetrix IDs, status flags, recovery SRDF groups, and SRDF mode in the output.

---

#### **Examples**

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 le> : 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

```

## symstar show command

### Description

The `symstar show` command displays the contents of the SRDF/Star definition file that was created by the `symstar setup` command.

### Note

To display all the devices with SRDF/Star, include the `-detail` option.

**Examples**

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
```

## symstar list command

### Description

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.

### Examples

To list the configurations for all the SRDF/Star composite groups, enter:

```
symstar list
```

#### S T A R   G R O U P S

| Name          | Flags<br>MLC | Workload<br>Name | Star<br>State | First Target |        | Second Target |        |
|---------------|--------------|------------------|---------------|--------------|--------|---------------|--------|
|               |              |                  |               | 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

### Note

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
- Cleaning up metadata

## Isolate SRDF/Star sites

### Description

There may be occasions when it is necessary to isolate one of the SRDF/Star sites, perhaps for testing purposes, and then rejoin the isolated site with the SRDF/Star configuration.

### 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.

Issue the `symstar isolate` command to temporarily isolate one or all of the SRDF/Star sites. 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 synchronous site.

## Isolate a protected target site

### Description

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.

### Examples

To isolate site **London** by splitting its SRDF pairs and making the R2 devices read/write-enabled to the **London** host:

```
symstar -cg StarGrp disable
symstar -cg StarGrp isolate -site London
```

## Isolate a disconnected target site

### Description

If the site you want to isolate is in the Disconnected state, first get it to the Protected state with the `connect` and `protect` commands.

### Examples

```
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site London
symstar -cg StarGrp isolate -site London
```

## Rejoin an isolated site

After performing testing or other tasks in **London** that require the isolation, rejoin the **London** site with the SRDF/Star configuration and enable SRDF/Star protection again. To

do this, first transition **London** from the Isolated state to the Disconnected state. Then proceed to connect and protect.

After rejoining the London site, reestablish any **London** BCV pairs that are part of the **StarGrp** composite group.

### Examples

```
symstar -cg StarGrp disconnect -site London
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

## Unprotect target sites

### Description

To unprotect the target sites, first turn off SRDF/Star protection (assuming the system state is STAR Protected).

### Options

#### **disable**

Disables SRDF/Star protection and terminates the SDDF sessions.

#### **unprotect**

Disables SRDF consistency protection and sets the STAR mode indicators.

### Example

Execute the following command sequence from the workload site (**NewYork**):

```
symstar -cg StarGrp disable
symstar -cg StarGrp unprotect -site NewJersey
symstar -cg StarGrp unprotect -site London
```

## Halt target sites

### Description

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.

#### **NOTICE**

All RDF links between the 3 sites, including the RDF links for the recovery leg, must be online before you initiate the halt operation.

### Examples

To halt SRDF/Star, enter:

```
symstar -cg StarGrp halt
```

## Clean up metadata

### Description

The **symstar cleanup** command cleans up internal metadata and array cache after a failure.

The cleanup action applies only to the asynchronous site.

**Examples**

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

The following configurations allow for dynamically adding or removing devices from an SRDF/Star consistency group while maintaining consistency protection if the group is in the Connected, Protected, or STAR-enabled states:

- Concurrent SRDF/Star CG
- Concurrent SRDF/Star CG with R22 devices
- Cascaded SRDF/Star CG
- Cascaded SRDF/Star CG with R22 devices

In SRDF/Star configurations, the `symstar modifycg` command with the `add` and `remove` options performs dynamic modification of SRDF/Star consistency groups.

### NOTICE

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 (`storrdpd`) 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.

#### NOTICE

Do not enable the `gns_remote_mirror` option in the GNS daemon's options file when using GNS with SRDF/Star. This option is not supported in SRDF/Star environments.

`gns_remote_mirror` 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, issue the `symstar buildcg` command to build a definition of the CG at each site in the SRDF/Star configuration.

## SRDF/Star consistency group restrictions

These restrictions apply to the `add` and `remove` options of the `symstar modifycg` command:

- Engenuity 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 Engenuity 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.

## Prepare staging for SRDF/Star consistency group modification

Before dynamically modifying SRDF/Star consistency groups, 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

The restrictions described in this section are in addition to the following:

- [SRDF/Star restrictions on page 250](#)
- [Restrictions: SRDF groups and devices for dynamic add operations on page 199](#)

The following additional restrictions apply to the SRDF groups and devices in the staging area for dynamic `symstar modifycg add` operations:

- Staging area cannot be an SRDF/Metro configuration.
- 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 SRDF 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 SRDF/Star configuration being updated:
  - 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.

## Add devices to a concurrent SRDF/Star consistency group

### Description

The `symstar modifycg` command moves devices between the staging area and the SRDF/Star CG, and updates the CG definition.



## 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
```

## Options

**-devs** *SymDevStart:SymDevEnd* or *SymDevName, SymDevStart:SymDevEnd* or *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 **-stg\_rdfg** 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 **-cg\_rdfg** 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.

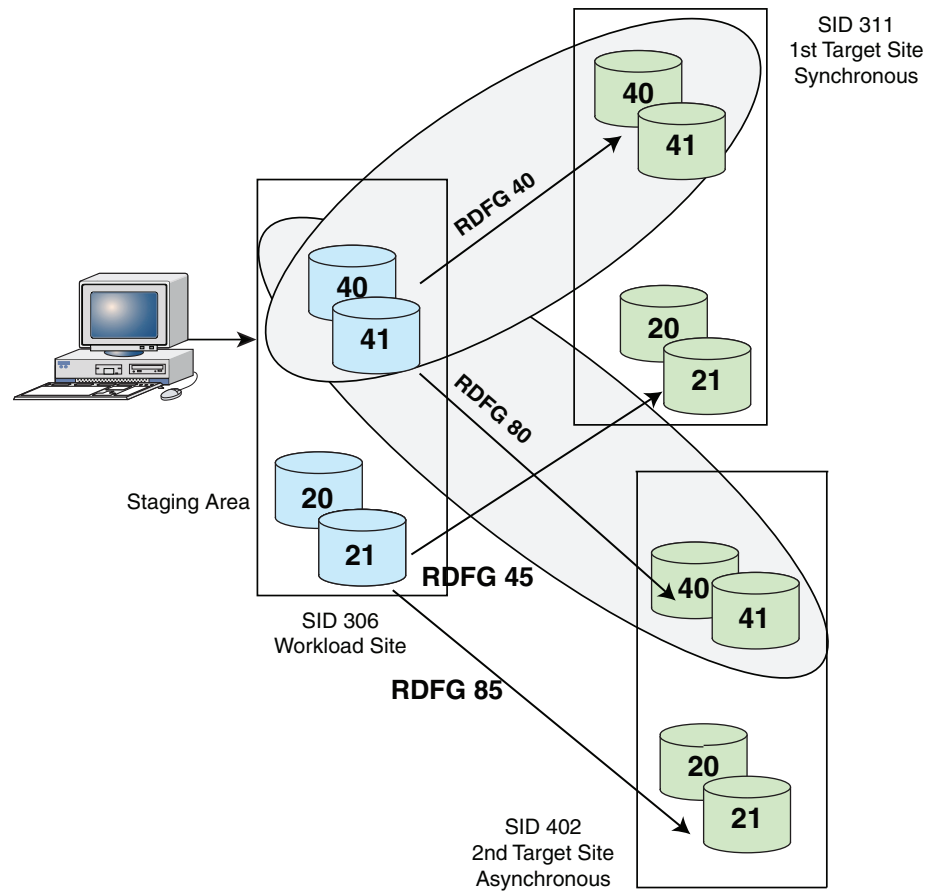
**-cg\_r21\_rdfg** *CgR21GrpNum*

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.

## Examples

The following example shows:

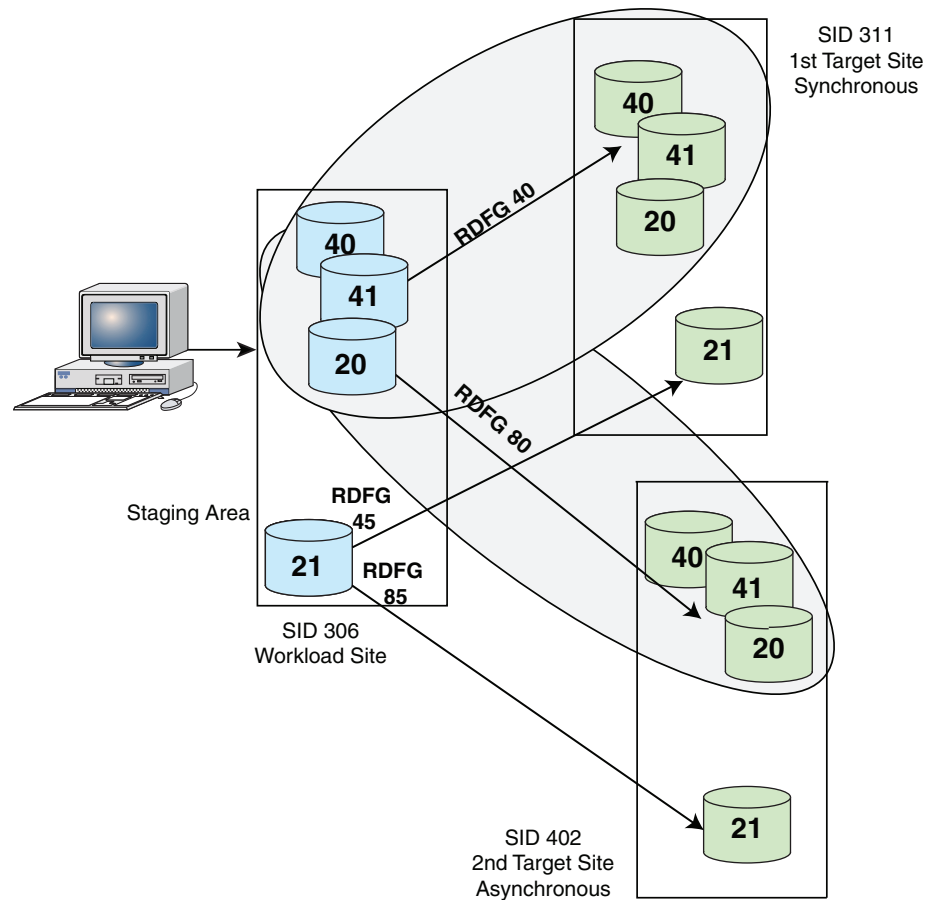
- CG **conStarCG** spans a concurrent SRDF/Star configuration.
- The 3 arrays are: 306, 311, and 402.
- The staging area contains devices 20 and 21.

**Figure 60** 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
```

The following image shows **ConStarCg** after device 20 was added. Note that device 21 is still in the staging area:

**Figure 61** ConStarCG after a dynamic add operation

### Restrictions

- 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/Star 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.
- The following table lists the valid SRDF/Star states for adding device pairs to a CG in a concurrent SRDF/Star configuration.

**Table 33** Allowable SRDF/Star states for adding device pairs to a concurrent CG

| State of 1st target site<br>(Synchronous) | State of 2nd target site<br>(Asynchronous) | STAR state  |
|-------------------------------------------|--------------------------------------------|-------------|
| Connected                                 | Connected                                  | Unprotected |
| Protected                                 | Connected                                  | Unprotected |
| Connected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Protected   |

## Verify moved devices in concurrent CG

### Description

Use the `symstar show -cg CgName -detail` command to check that the devices were moved to the concurrent CG.

### Example

To check if device 20 was added to `ConStarCG`:

```
symstar show -cg ConStarCG -detail
```

## Add devices to a cascaded SRDF/Star consistency group

The `symstar -cg CgName modifycg -add` command moves the devices from the staging area to the SRDF group(s).

### Restrictions

The following table shows the valid states for adding device pairs to a CG in a cascaded SRDF/Star configuration.

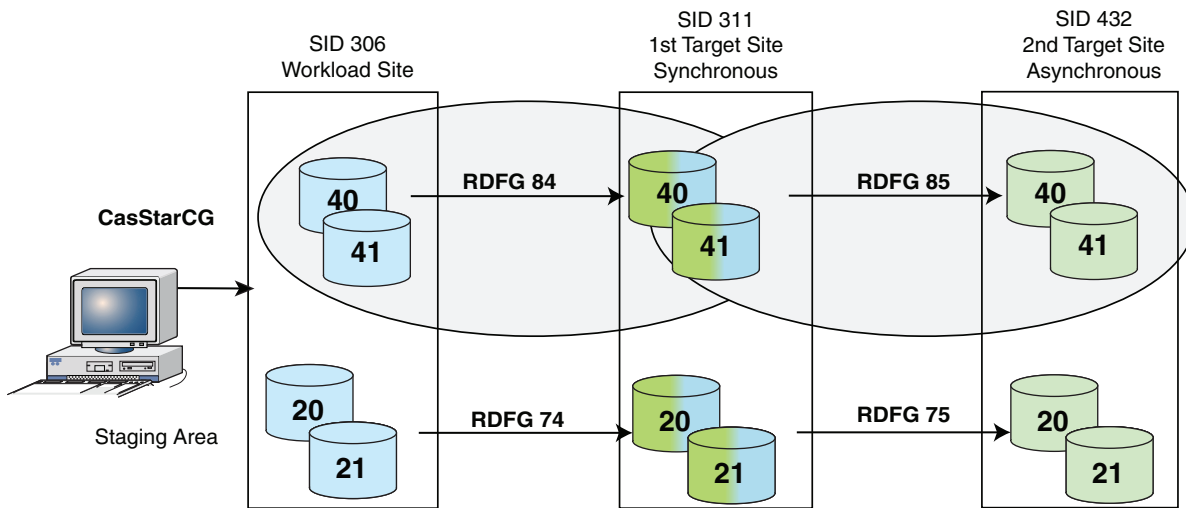
**Table 34** Allowable states for adding device pairs to a cascaded CG

| State of 1st target site<br>(Synchronous) | State of 2nd target site<br>(Asynchronous) | STAR state  |
|-------------------------------------------|--------------------------------------------|-------------|
| Connected                                 | Connected                                  | Unprotected |
| Protected                                 | Connected                                  | Unprotected |
| Protected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Protected   |

### Example

The following example shows:

- CG `CasStarCG` spans a cascaded SRDF/Star configuration.
- The 3 arrays are: 306, 311, and 402.
- The staging area contains devices 20 and 21.

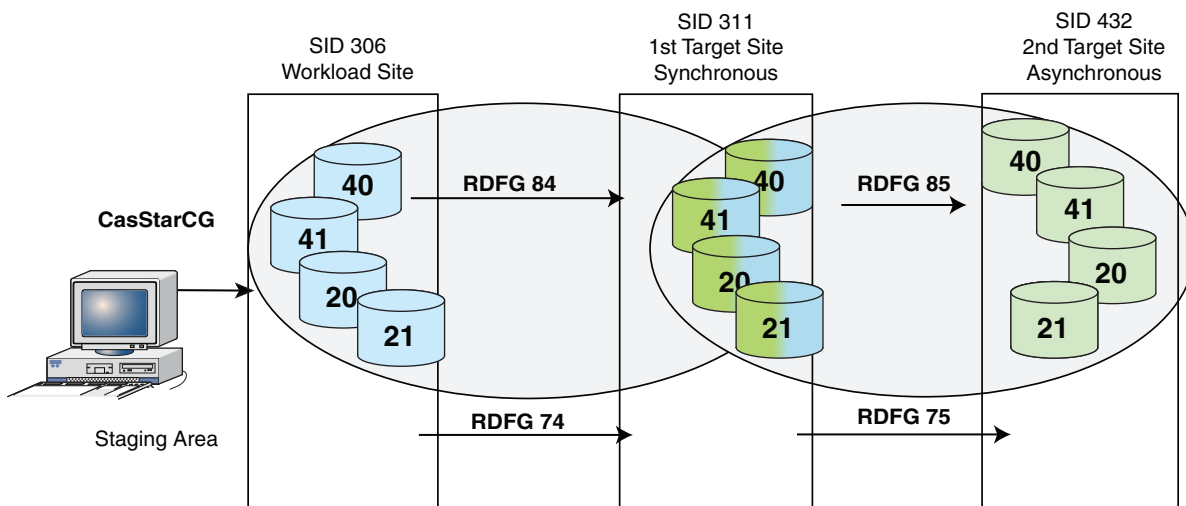
**Figure 62** Adding devices to a cascaded SRDF/Star CG

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
```

The following image 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 63** CasStarCG after a dynamic add operation

#### Pair states of devices in a CG after symstar modifycg -add

The following table 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 35** 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 ( <code>-cons_exempt</code> ) 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.                                                                           |

## Verifying moved devices in cascaded CG

### Description

Use the `symstar show -cg CgName -detail` command to verify that the devices were moved.

### Examples

To verify devices 20 and 21 were added to **CasStarCG**:

```
symstar show -cg CasStarCG -detail
```

## Remove 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

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.

## Remove devices from an SRDF/Star concurrent consistency group

### 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
```

### Restrictions

The following table shows the valid states for removing device pairs from a CG in a concurrent SRDF/Star configuration.

**Table 36** Allowable states for removing device pairs from a concurrent SRDF/Star CG

| State of 1st target site<br>(Synchronous) | State of 2nd target site<br>(Asynchronous) | Star state  |
|-------------------------------------------|--------------------------------------------|-------------|
| Connected                                 | Connected                                  | Unprotected |
| Protected                                 | Connected                                  | Unprotected |
| Connected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Protected   |

### Verify remove operation for concurrent CG

#### Example

To check if the dynamic remove operation was successful for ConStarCG:

```
symstar show -cg ConStarCG -detail
```

### Remove devices from an SRDF/Star cascaded consistency group

#### 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
```

### Restrictions

The following table shows the valid states for removing device pairs from a CG in a cascaded SRDF configuration.

**Table 37** Allowable states for removing device pairs from a cascaded SRDF/Star CG

| State of 1st target site<br>(Synchronous) | State of 2nd target site<br>(Asynchronous) | Star state  |
|-------------------------------------------|--------------------------------------------|-------------|
| Connected                                 | Connected                                  | Unprotected |
| Protected                                 | Connected                                  | Unprotected |
| Protected                                 | Protected                                  | Unprotected |
| Protected                                 | Protected                                  | Protected   |

## Verify remove operation for cascaded CG

### Example

To check if the dynamic remove operation was successful for ConStarCG:

```
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:

### Procedure

1. Reissue the `modifycg` command using *exactly* the same 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, specify `-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.

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, issue the `symstar -cg CgName query -detail` command to display whether the Composite Group State is `RecoveryRequired`.  
To display CG **SampleCG**:

```
symstar -cg SampleGCG query -detail
```

2. Issue the `symstar -cg CgName recover -force` command to retry the failed operation.  
To retry 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

The following table 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 38** 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 <sup>a</sup>                       |
| PathFail                 | SRDF/S                     | Suspended <sup>a</sup>                       |
| PathFail                 | SRDF/A                     | Suspended <sup>a</sup>                       |

a. The SRDF pair state can be Partitioned instead of Suspended 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, reissue the `symstar modifycg` operation.

# Recovery operations: Concurrent SRDF/Star

This section describes Concurrent SRDF/Star recovery from transient faults with or without reconfiguration.

## Recover 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.

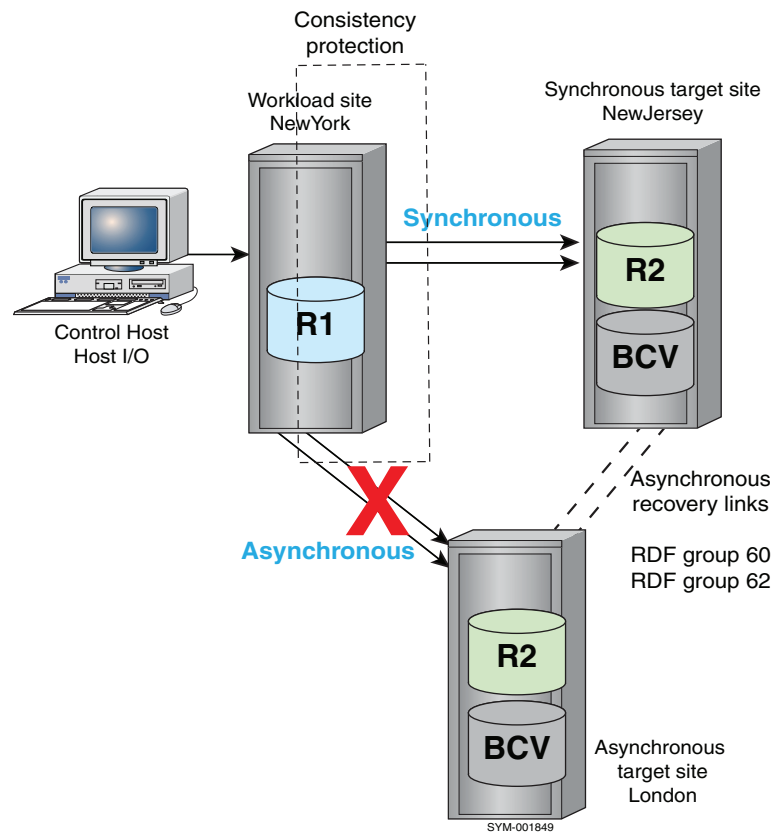
An SRDF/Star fault caused by network or remote storage controller faults is a transient fault.

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.

The following image shows a temporary interruption on the SRDF/A link in a concurrent SRDF/Star environment:

**Figure 64** Transient failure: concurrent SRDF/Star



There are two methods to clean up and restore SRDF/Star:

- When the transient fault is corrected, clean up the internal metadata and the cache at the asynchronous target site and return the site to SRDF/Star Protected. [Recover from a transient fault without reconfiguration: concurrent SRDF/Star on page 298](#) 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, reconfigure SRDF/Star to recover the asynchronous site. [Recover from transient fault with reconfiguration: concurrent SRDF/Star on page 299](#) describes the steps to avoid a long wait when the asynchronous site must be recovered sooner than the transient fault will be repaired.

## Recover from a transient fault without reconfiguration: concurrent SRDF/Star

If the synchronous target (New Jersey in [Figure 65 on page 300](#)) state is Protected, and the asynchronous target (London) state is PathFail.

### Procedure

1. Issue the `symstar -cg CgName 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
```

**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 SRDF/Star data at the remote site are protected with TimeFinder BCVs:

- If SRDF/Star data at the remote site are protected with TimeFinder BCVs, proceed to Step 2.
  - If not, skip to Step 3.
2. If SRDF/Star data at the remote site are protected with TimeFinder BCVs, 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. Issue the `symstar -cg CgName` 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 site `London`:

```
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, issue the `symmir -cg CgName establish` command to reestablish them.

To reestablish the BCV pairs:

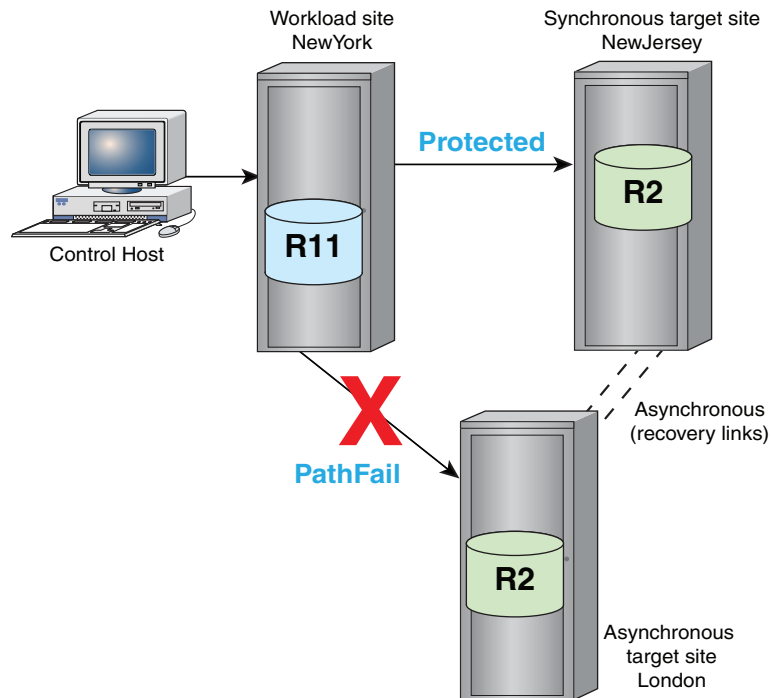
```
symmir -cg StarGrp establish -star -rdf -rdfg name:London
```

## Recover 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 SRDF/Star protection.

The following procedure describes the steps to recover 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 65** Transient fault recovery: before reconfiguration

The image 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.

## Recover using reconfigure operations

Use the `reconfigure` operation (to change the mode to Cascaded SRDF/Star) as the initial recovery step.

### 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
```

### Options

- 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.

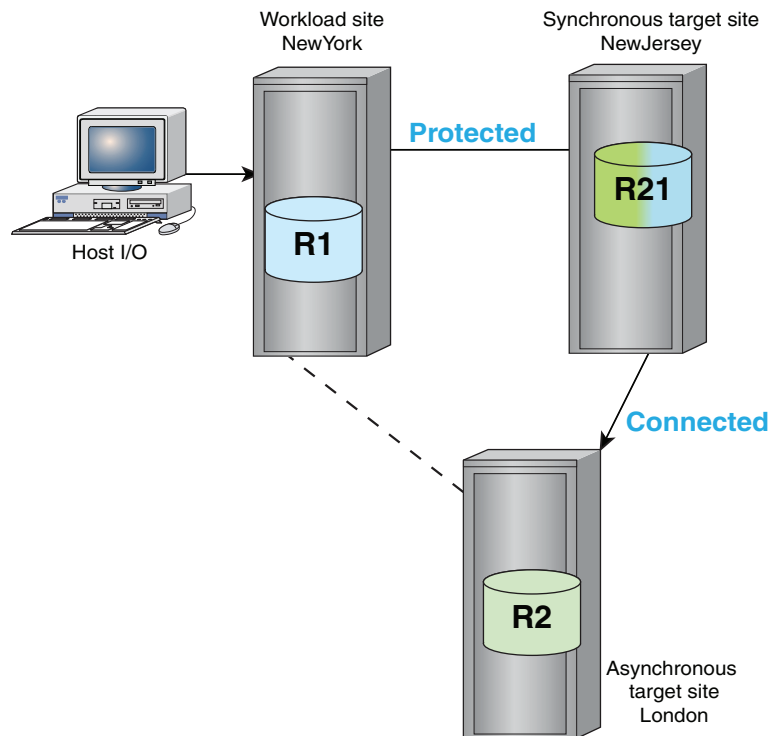
### Example

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 66** Transient fault recovery: after reconfiguration



### Restrictions

- If the asynchronous target site is in the Disconnected state and STAR is unprotected, specify the `-full`.
- If the asynchronous target site is in the PathFail state and STAR is unprotected, specify the `-reset` and `-full` options.
- Specify the `-full` option only when an SRDF incremental resynchronization is not available.
- Perform the `recover` operation to recover from PathFail (asynchronous target site) and a tripped state (SRDF/Star).

## Workload switching: Concurrent SRDF/Star

This section describes the following topics for a Concurrent SRDF/Star configuration:

- Planned workload switching
- Unplanned workload switching to synchronous or asynchronous target site

- Switch back to the original workload site

## 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.

### NOTICE

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:

### Procedure

1. Confirm the system state using the `symstar query` command.
2. Stop the application workload at the current workload site, unmount the file systems, and export the volume groups.
3. Perform the SRDF/Star `halt` action from the control host.

To halt CG StarGrp:

```
symstar -cg StarGrp halt
```

---

### Note

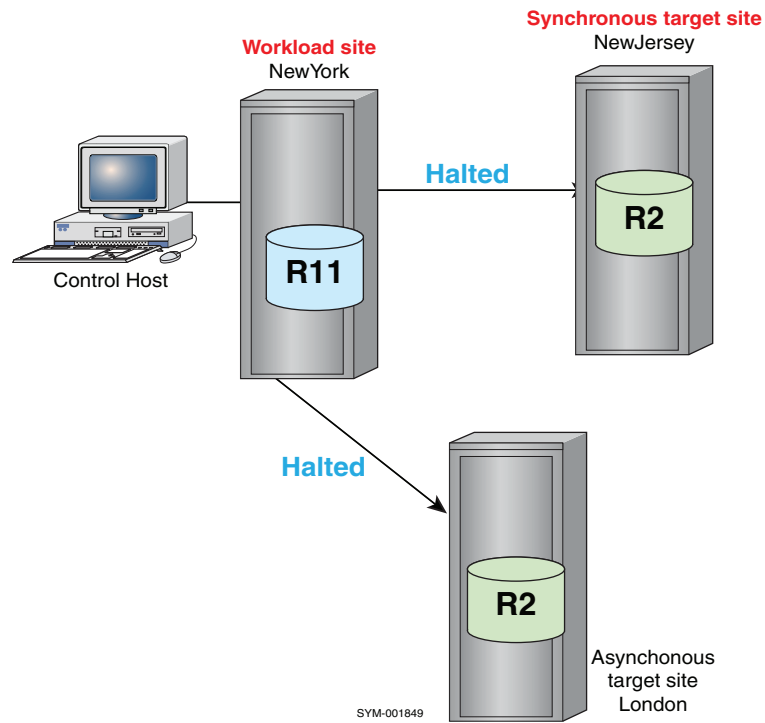
If you change your mind after halting SRDF/Star, issue the `halt -reset` command to restart the workload site on the same host.

---

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.

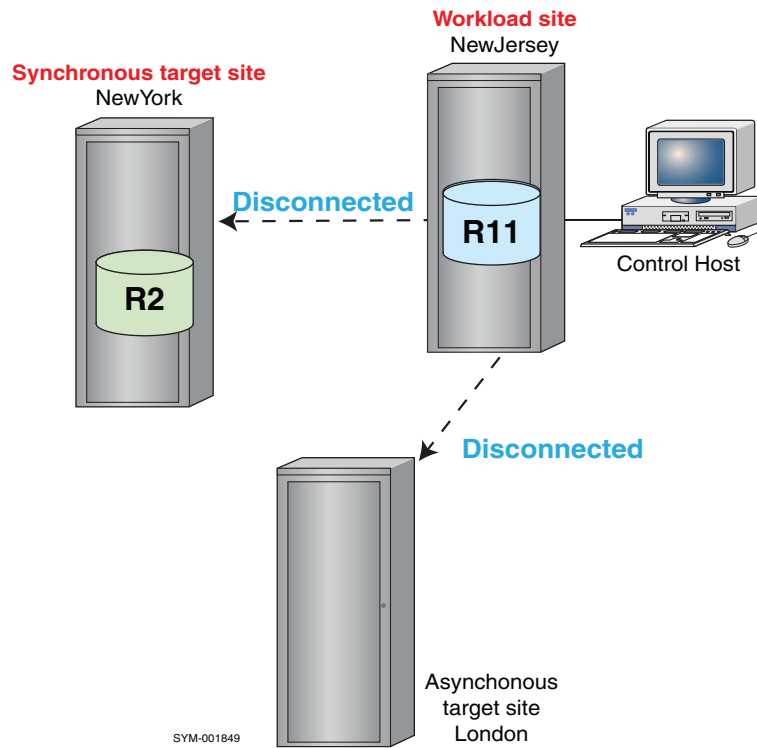
The target sites transition to the Halted state, with all three sites having the data.

**Figure 67** Concurrent SRDF/Star: halted

4. From a control host at the synchronous target site (NewJersey), issue the switch command to switch the workload to the synchronous target site (NewJersey).

```
symstar -cg StarGrp switch -site NewJersey
```

The following image shows the resulting SRDF/Star state:

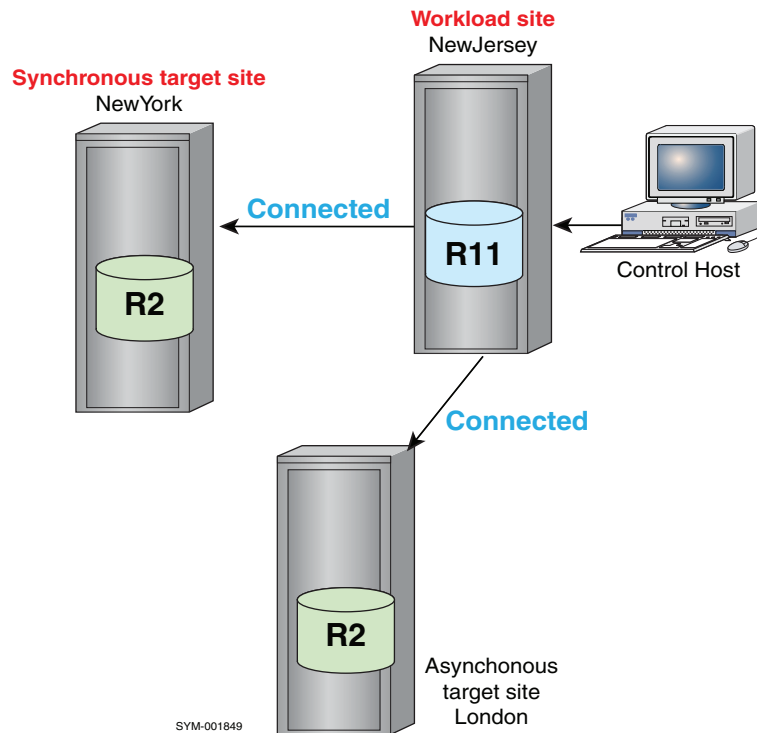
**Figure 68** Concurrent SRDF/Star: switched

5. From a control host at the synchronous target site (NewJersey), issue 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 following image shows the resulting SRDF/Star state:



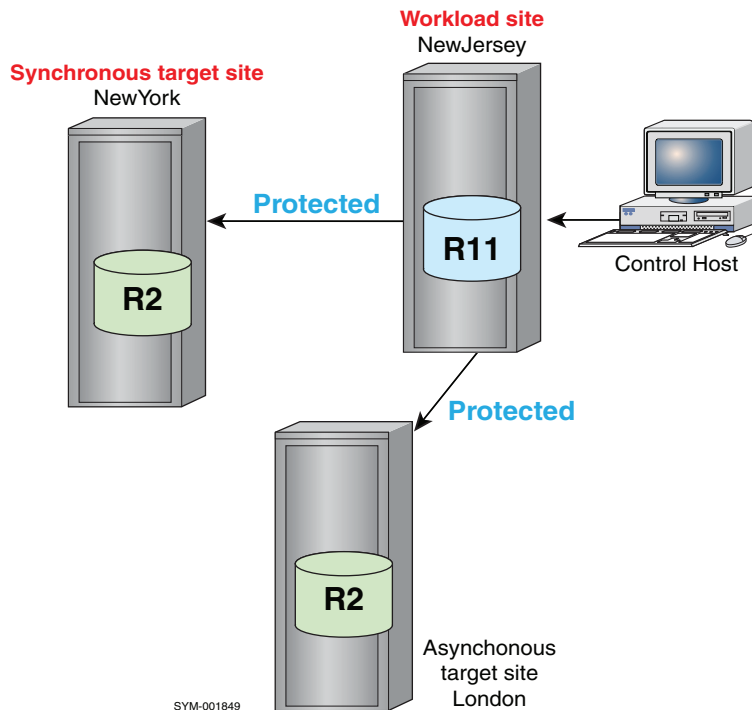
**Figure 69** Concurrent SRDF/Star: connected

6. From a control host at the synchronous target site (NewJersey), issue 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 following image shows the resulting SRDF/Star state:

**Figure 70** Concurrent SRDF/Star: protected

## Unplanned workload switching: concurrent SRDF/Star

Loss of the workload site (NewYork) is a disaster because it disrupts the workload.

Issue the `switch` command to:

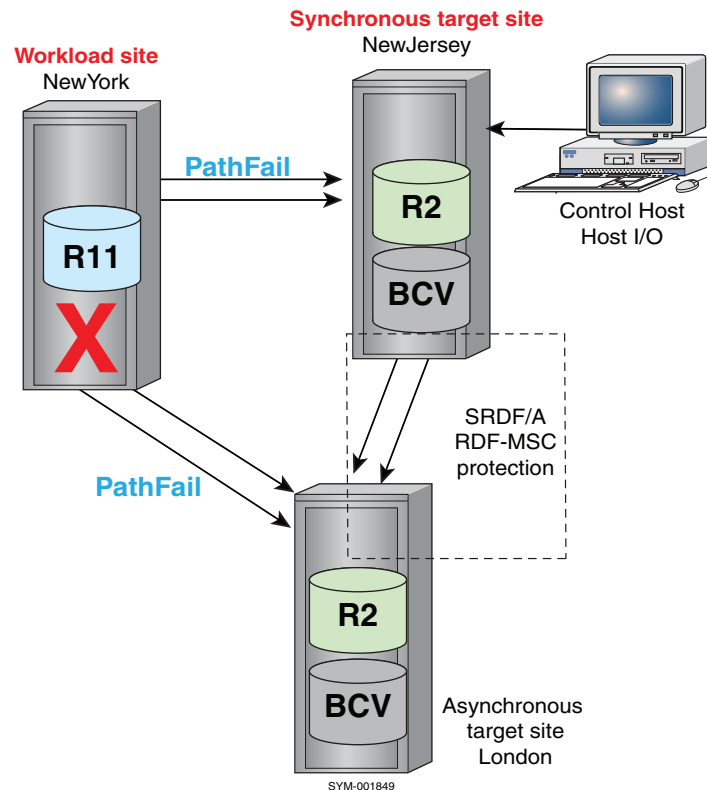
- Switch the workload to either one of the remote sites, and
- Resume data replication

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.

The following image shows concurrent SRDF/Star where a disaster fault has caused the loss of the workload site (NewYork):

**Figure 71** 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, issue the following command to get the asynchronous site to the *PathFail* state:

```
symstar -cg StarGrp disconnect -trip -site London
```

### Procedure

1. From a control host at the synchronous target site (*NewJersey*), issue 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 SRDF/Star data at the remote site are protected with TimeFinder BCVs:

- If SRDF/Star data at the remote site are protected with TimeFinder BCVs, proceed to Step 2.
  - If not, skip to Step 3.
2. If SRDF/Star data are protected 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`.

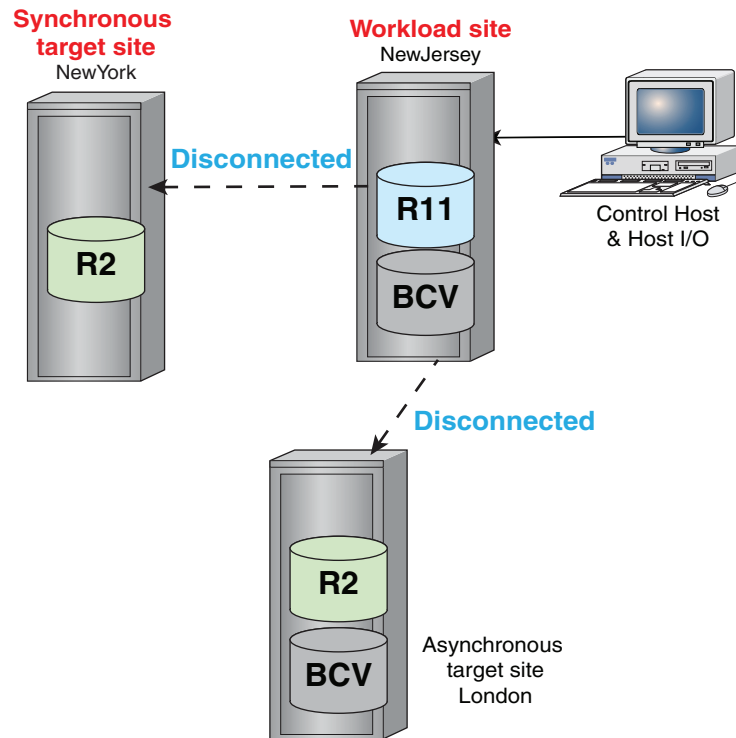
To split off a consistent restartable image of the data volumes during the resynchronization process using the device file:

```
symmir -f StarFileLondon split -star -sid 13
```

3. From a control host at the synchronous target site (`NewJersey`), issue 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
```

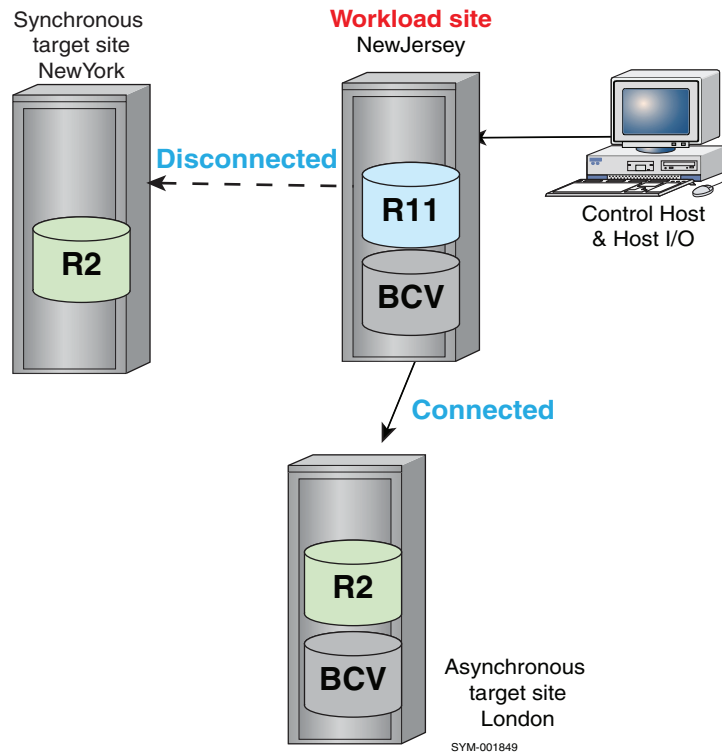
The following image shows the resulting SRDF/Star state:

**Figure 72** Concurrent SRDF/Star: workload switched to synchronous site

- From a control host at the synchronous target site (NewJersey), issue the `connect` command to connect NewJersey to London (asynchronously):

```
symstar -cg StarGrp connect -site London
```

The following image shows the resulting SRDF/Star state:

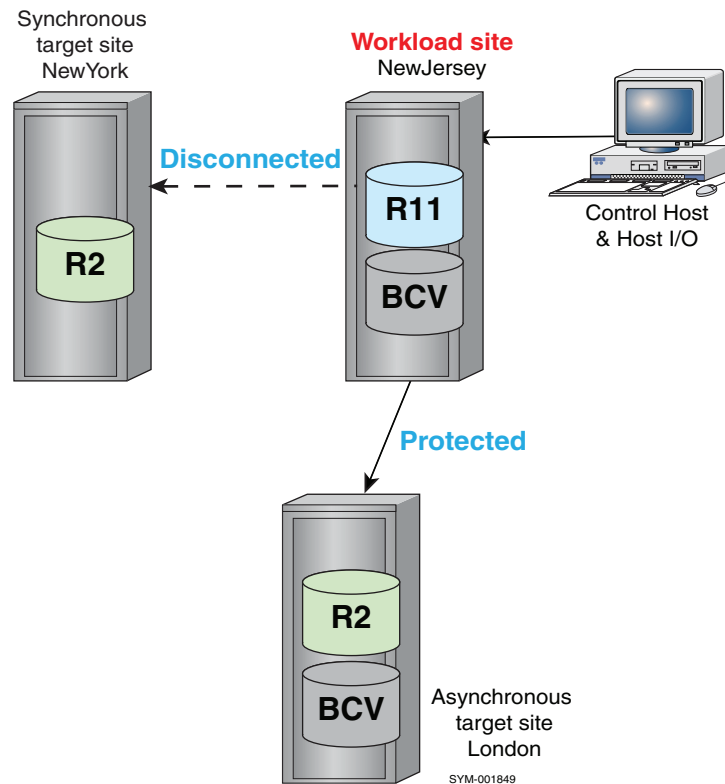
**Figure 73** Concurrent SRDF/Star: new workload site connected to asynchronous site

5. From a control host at the synchronous target site (NewJersey), issue the `protect` and `enable` commands to:

- `Protect NewJersey to London`
- `Enable SRDF/Star`

```
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The following image shows the resulting SRDF/Star state:

**Figure 74** 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 SRDF/Star data at the remote site are protected with TimeFinder BCVs:

- If RDF/Star data at the remote site are protected with TimeFinder BCVs, proceed to Step 6.
- If not, 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
```

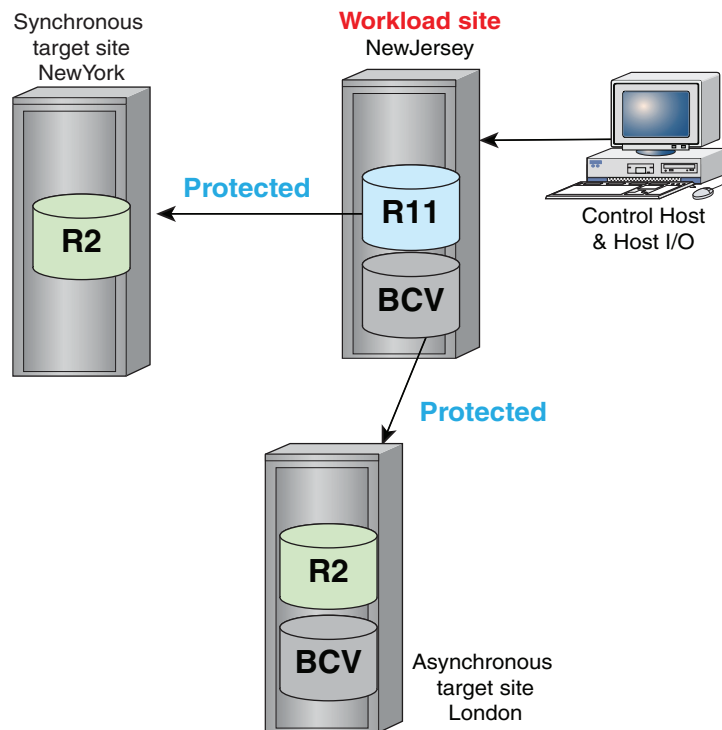
- When the NewYork site is repaired, you may want to bring NewYork back into the SRDF/Star while retaining the workload site at NewJersey.

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 following image shows the resulting SRDF/Star state:

**Figure 75** 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`).

### Procedure

1. From a control host at the asynchronous target site (`London`), issue 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 SRDF/Star data at the remote site are protected with TimeFinder BCVs:

- If SRDF/Star data at the remote site are protected with TimeFinder BCVs, proceed to Step 2.
  - If not, skip to Step 3.
2. If SRDF/Star data are protected 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`), issue 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.

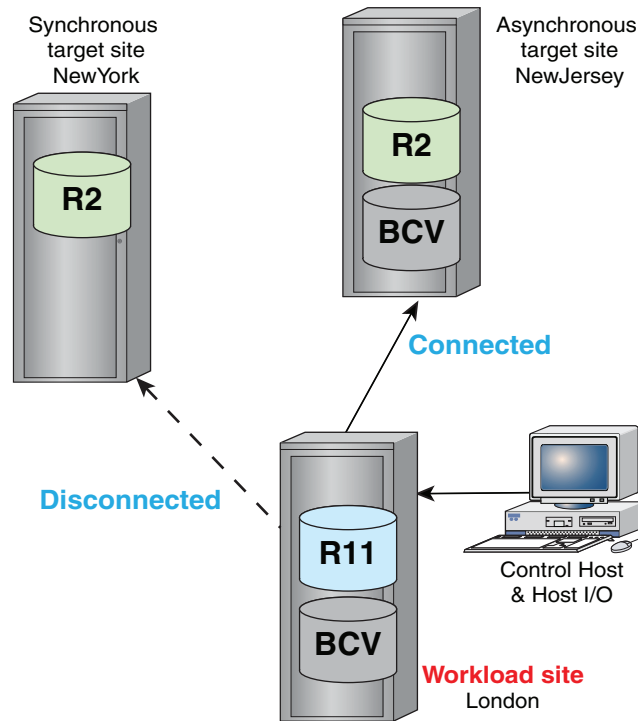
The `London` site connects to the `NewJersey` site and retrieves the `NewJersey` data.

**Note**

The `connect` action is not required because the `switch` action specified that SRDF retrieve the remote data from the `NewJersey` site.

The following image shows the resulting SRDF/Star state:

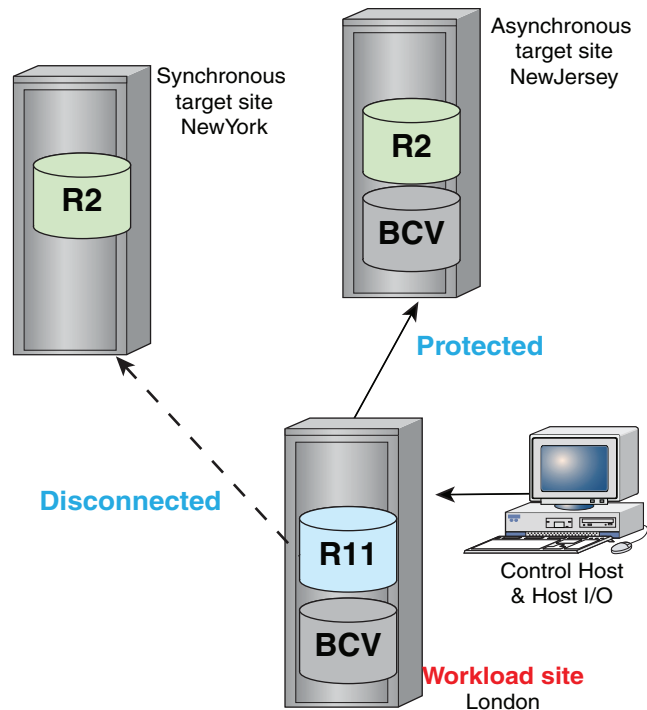
**Figure 76** Concurrent SRDF/Star: workload switched to asynchronous site



- From a control host at the asynchronous target site (London), issue the `protect` command to protect London to NewJersey:

```
symstar -cg StarGrp protect -site NewJersey
```

The following image shows the resulting SRDF/Star state:

**Figure 77** 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 SRDF/Star data at the remote site are protected with TimeFinder BCVs:

- If SRDF/Star data at the remote site are protected with TimeFinder BCVs, proceed to Step 5.
- If not, 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 -rdrg 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`).

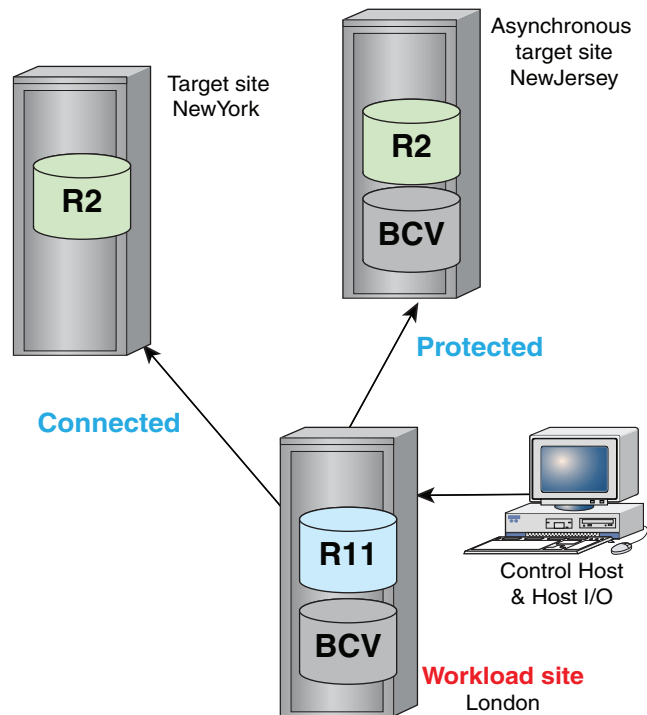
However, you can connect to `NewYork`. The `connect` action sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links.

To connect to `NewYork`, issue the `connect` command from the `London` site:

```
symstar -cg StarGrp connect -site NewYork
```

The following image shows the resulting SRDF/Star state:

**Figure 78** 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, 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 switching back to the original workload site vary depending on whether the workload was switched to the synchronous target site or the asynchronous target site:

- When switched 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
- When switched 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 75 on page 312](#).

### Procedure

1. Stop the workload at the site where the control host is connected.
2. Issue 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 operations: Cascaded SRDF/Star

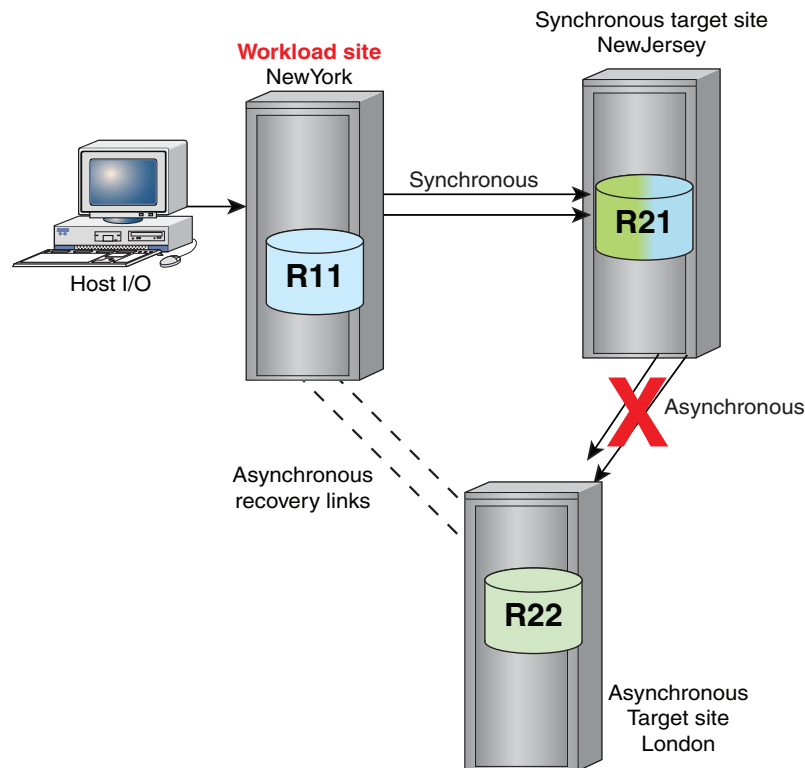
This section describes the following topics for a Cascaded SRDF/Star configuration:

- Recovering from transient faults without reconfiguration
- Recovering from transient faults with reconfiguration

### Recovering from transient faults: Cascaded SRDF/Star

The following image shows a temporary interruption (transient fault) on the SRDF/A link in a cascaded SRDF/Star environment:

**Figure 79** Transient fault: cascaded SRDF/Star



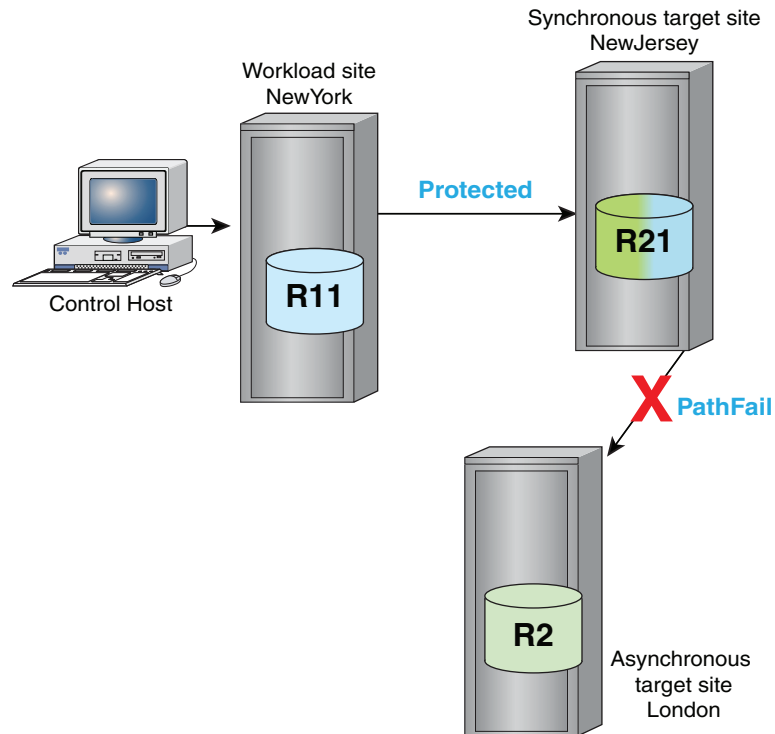
There are two methods to clean up and restore SRDF/Star:

- When the transient fault is corrected, 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 319](#) 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, reconfigure SRDF/Star to recover the asynchronous site. [Recovering from transient faults with reconfiguration: Cascaded SRDF/Star on page 320](#) 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

The following image shows the SRDF states when links to the asynchronous target site are down:

**Figure 80** Cascaded SRDF/Star with transient fault



The SRDF devices are now in the Suspended state.

### Procedure

1. Display the state of SRDF devices and the SRDF links that connect them using the `symrdf list` command.

See [Table 6 on page 40](#) 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. Transition links to the asynchronous site to the PathFail state using the `symstar -cg CgName disconnect -trip` command.

```
symstar -cg StarGrp disconnect -trip -site London
```

3. Issue 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

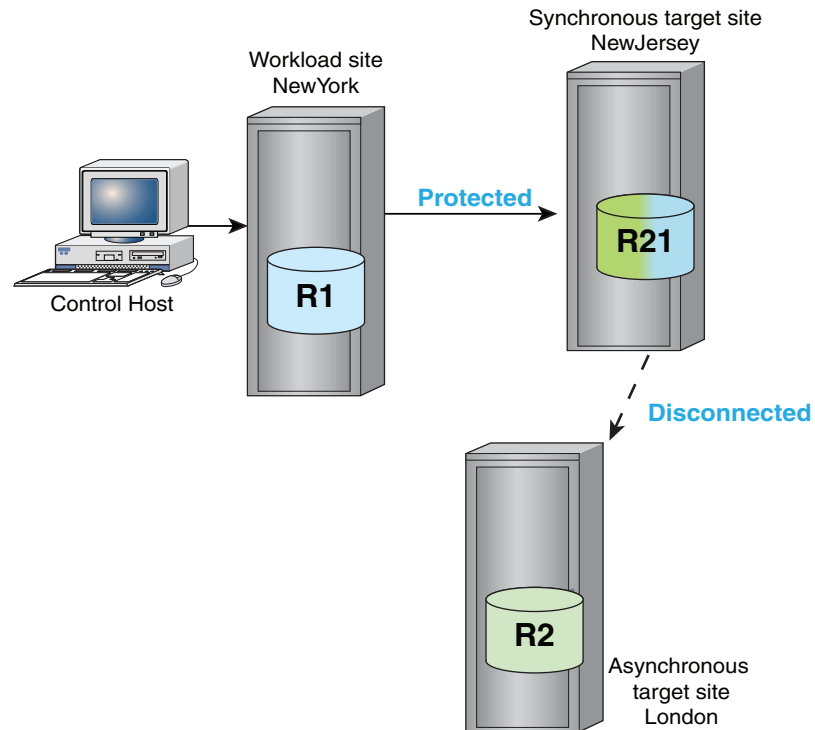
4. From the control host at the workload site, issue the `symstar -cg CgName reset` command to clean up any internal metadata or cache remaining at the asynchronous site after the transient fault occurred.

To clean up cache and metadata for CG `StarGrp` at site `London`:

```
symstar -cg StarGrp reset -site London
```

The following image shows the resulting SRDF/Star states:

**Figure 81** 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 80 on page 319](#) 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. Transition links to the asynchronous site to the `PathFail` state using the `symstar -cg CgName disconnect -trip` command.

```
symstar -cg StarGrp disconnect -trip -site London
```

2. Issue the `symstar reconfigure` command from the workload site (`NewYork`) control host.

See [Recover using reconfigure operations on page 300](#) and [Restrictions on page 301](#).

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
```

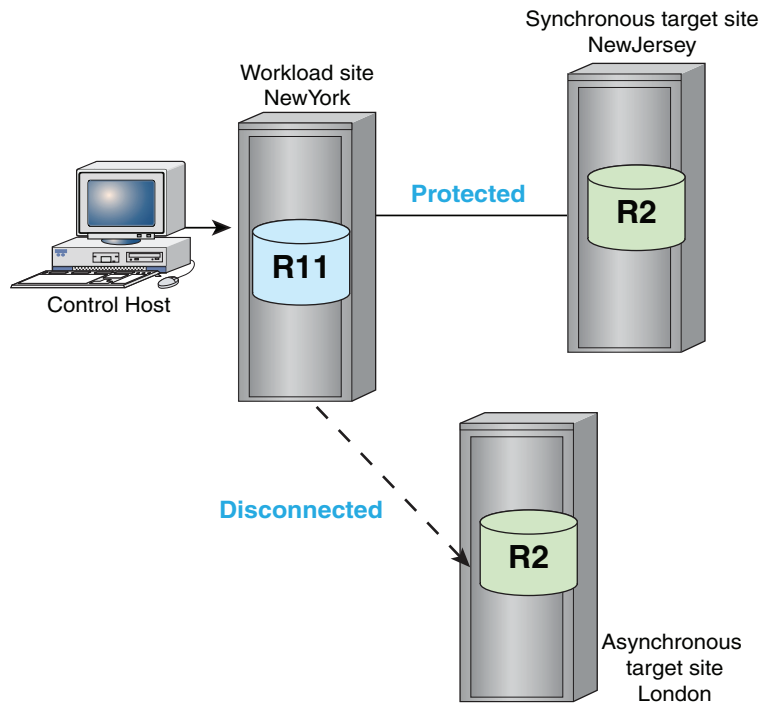
---

#### Note

If the system was not STAR Protected, specify the `-full` option to perform full resynchronization.

---

The following image shows the resulting SRDF/Star states:

**Figure 82** SRDF/Star: after reconfiguration to concurrent

## Workload switching: Cascaded SRDF/Star

This section describes the following topics for a Cascaded SRDF/Star configuration:

- Planned workload switching
- Unplanned workload switching to synchronous or asynchronous target site

### Planned workload switching: Cascaded SRDF/Star

Maintenance, testing and other activities may require switching 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.

When switching 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.

### Procedure

1. At the current workload site (NewYork), perform the SRDF/Star `halt` action.

To halt CG StarGrp:

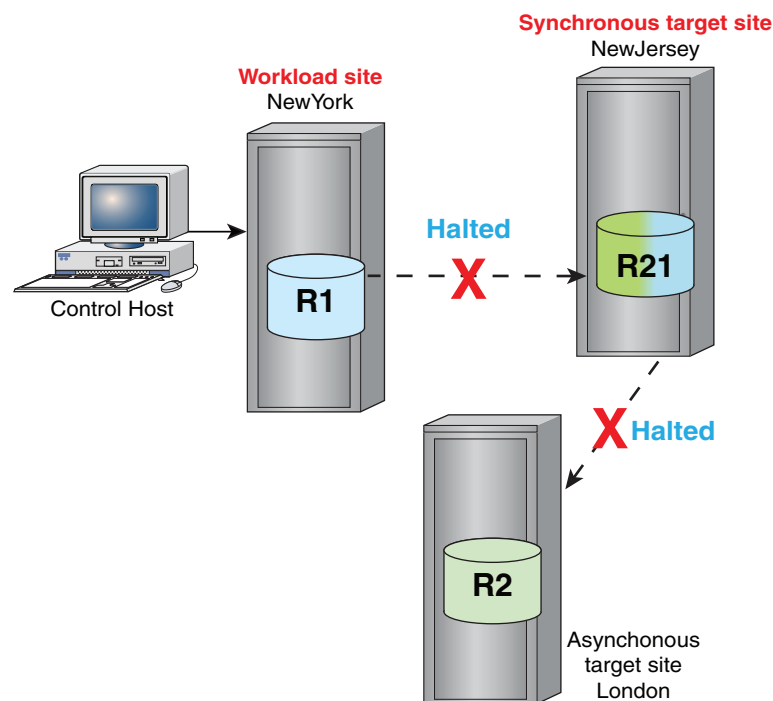
```
symstar -cg StarGrp halt
```

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.

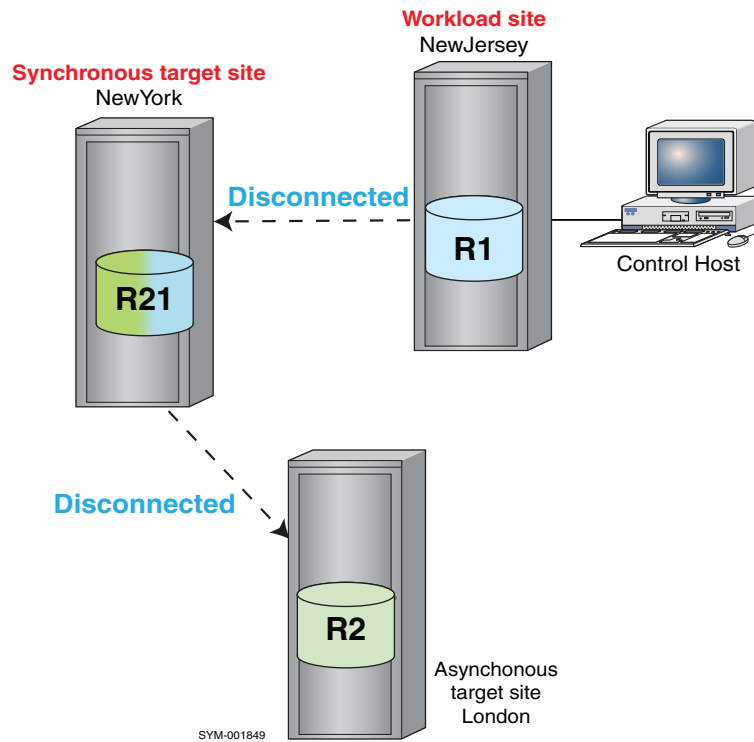
**Figure 83** Cascaded SRDF/Star: halted



2. From a control host at the synchronous target site (NewJersey), issue the `switch` command to switch the workload to the synchronous target site (NewJersey).

```
symstar -cg StarGrp switch -site NewJersey
```

The following image shows the resulting SRDF/Star state:

**Figure 84** 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

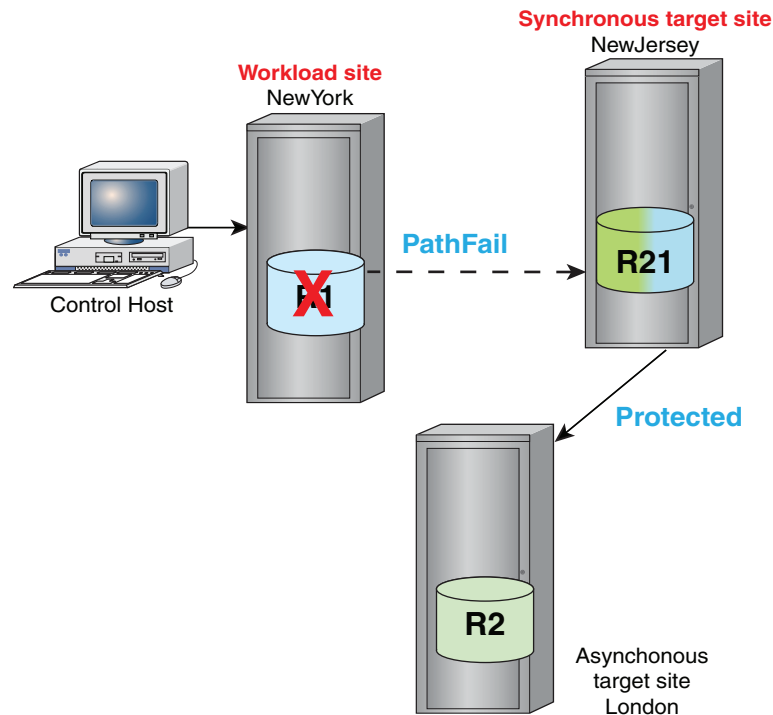
In cascaded mode, data at the synchronous target site is always more current than the data at asynchronous target site.

**NOTICE**

You cannot retain the data at the asynchronous target site if you move the workload to the synchronous target site.

In the following image, loss of the workload site (NewYork) has resulted in a system state of *NewJersey:Pathfail*:

**Figure 85** Loss of workload site: cascaded SRDF/Star



### Procedure

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), issue the `symstar cleanup` command to clean up any internal metadata or cache remaining at the asynchronous site.

To clean up the London site:

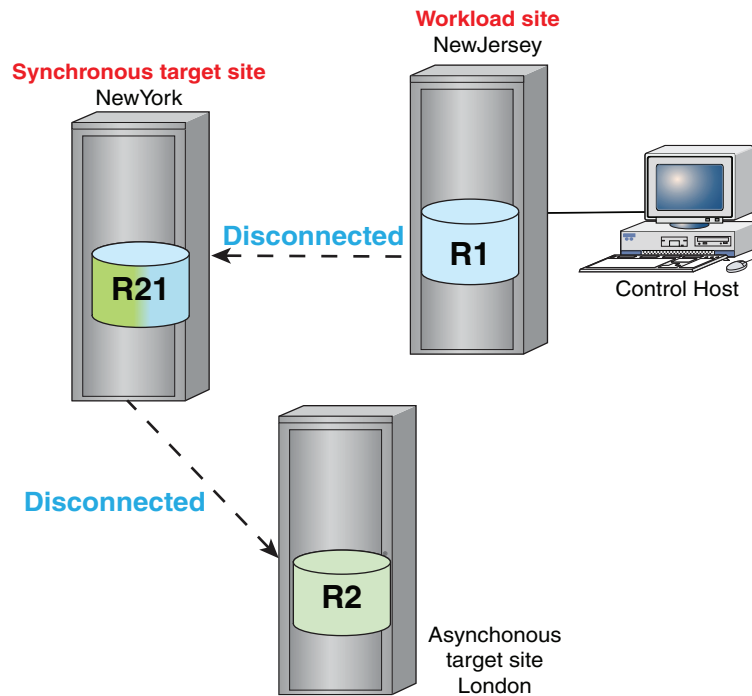
```
symstar -cg StarGrp cleanup -site London
```

3. From a control host at the synchronous target site (NewJersey), issue 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 following image shows the resulting SRDF/Star state:

**Figure 86** Workload switched to synchronous target site: cascaded SRDF/Star



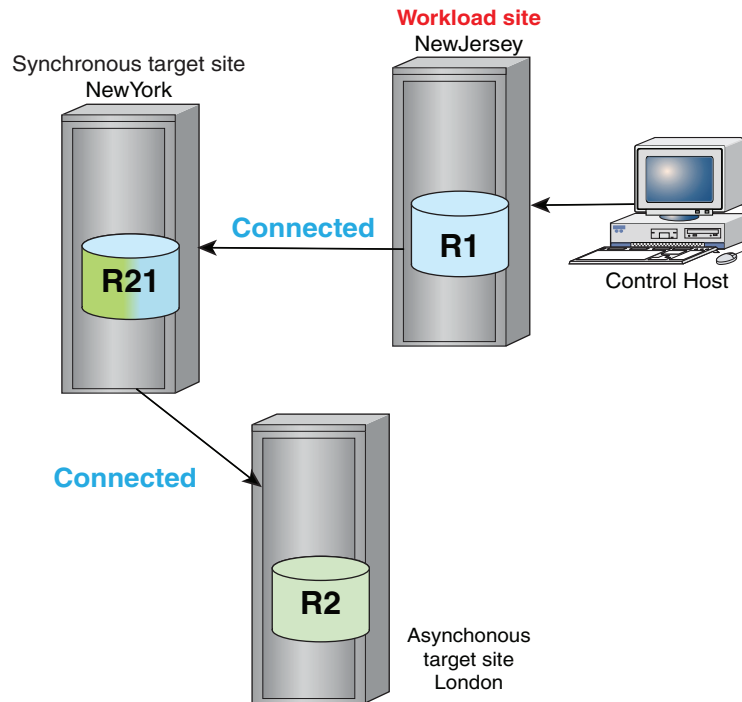
4. If data is protected with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

For details, see [Step 7: \(Optional\) Add BCV devices to the SRDF/Star configuration on page 278](#).

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**), issue two `connectconnect` 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 following image shows the resulting SRDF/Star state:

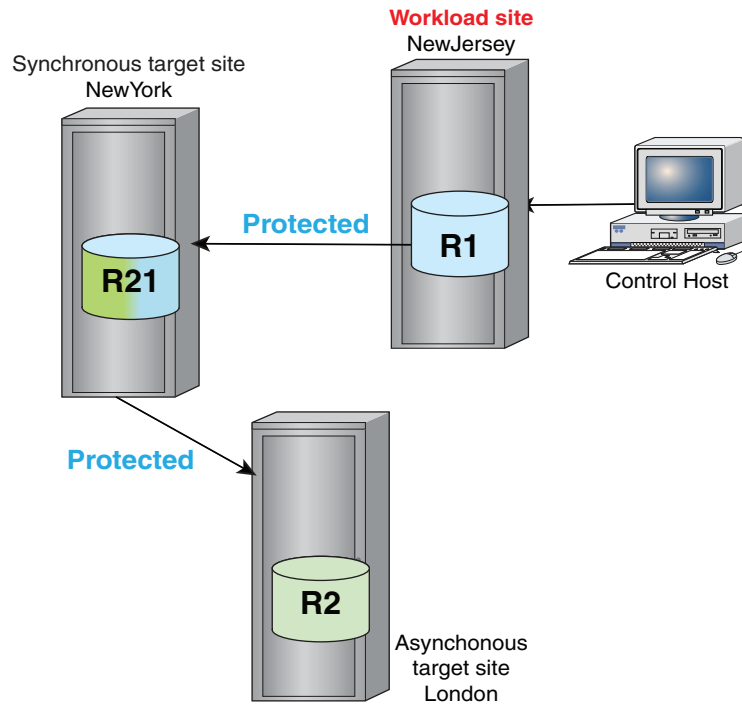
**Figure 87** After workload switch to synchronous site: cascaded SRDF/Star

7. From a control host at the new workload site (NewJersey), issue 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 following image shows the resulting SRDF/Star state:

**Figure 88** Cascaded SRDF/Star after workload switch: protected

8. From a control host at the new workload site, issue the `symstar reconfigure` command from the workload site to change the mode to concurrent.

See [Recover using reconfigure operations on page 300](#).

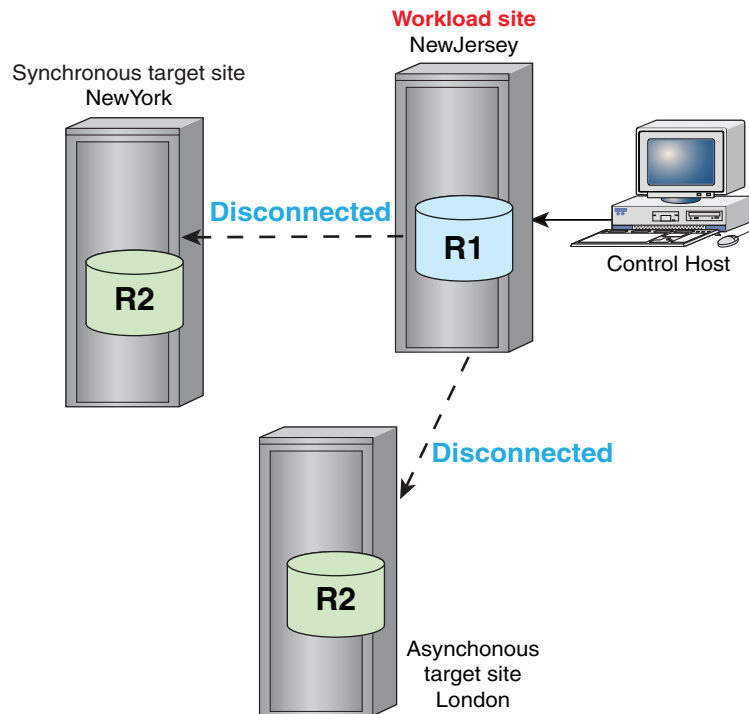
To reconfigure SRDF/Star to operate in concurrent mode with:

- 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 following image shows the resulting SRDF/Star configuration:



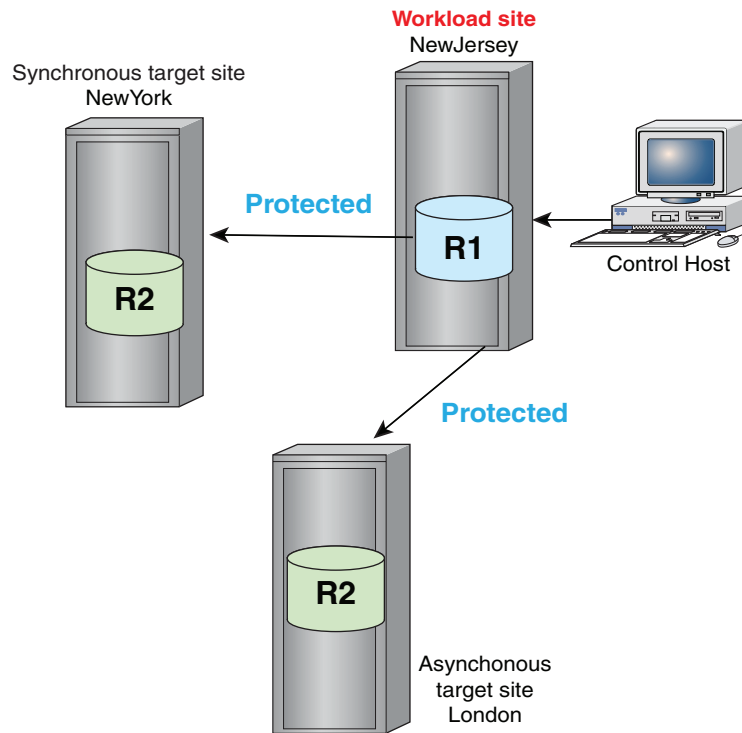
**Figure 89** After reconfiguration to concurrent mode

9. Run the following commands from a control host at the new workload 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 following image shows the resulting SRDF/Star configuration:

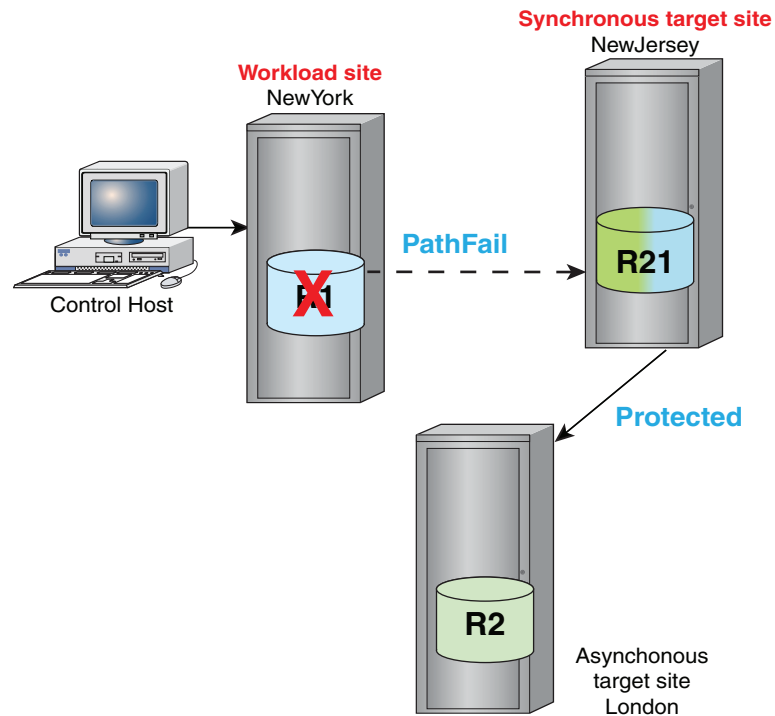
**Figure 90** Protected after reconfiguration from cascaded to concurrent mode

### Unplanned workload switching to asynchronous target site: Cascaded SRDF/Star

This section describes two procedures to switch the workload to the asynchronous target site and keep the synchronous or asynchronous site's data.

#### Switch workload site: keep asynchronous site's data

In the following image, the workload site (`NewYork`) has been lost:

**Figure 91** 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):

#### Procedure

1. If London is in a Protected state, issue the `disconnect` command:

```
symstar -cg StarGrp disconnect -trip -site London
```

2. If the `disconnect` leaves London in a CleanReq state, issue the `cleanup` command:

```
symstar -cg StarGrp cleanup -site London
```

3. Issue the `switch` command to switch the workload site to the asynchronous target site (London) and keep the asynchronous target's (London) data:

```
symstar switch -cg StarGrp -site London -keep_data London
```

4. The London site is at asynchronous distance from both New York and New Jersey. SRDF/Star supports only one asynchronous site.

When the New York site is repaired, you cannot connect and protect New York without switching the workload back to a configuration that has only one asynchronous site (New York or New Jersey).

However, you can connect to New York. The `connect` action sets the mode to adaptive copy disk and brings the devices to RW on the SRDF links.

Issue two `connect` commands to connect the workload site (London) to both target sites (NewJersey and NewYork):

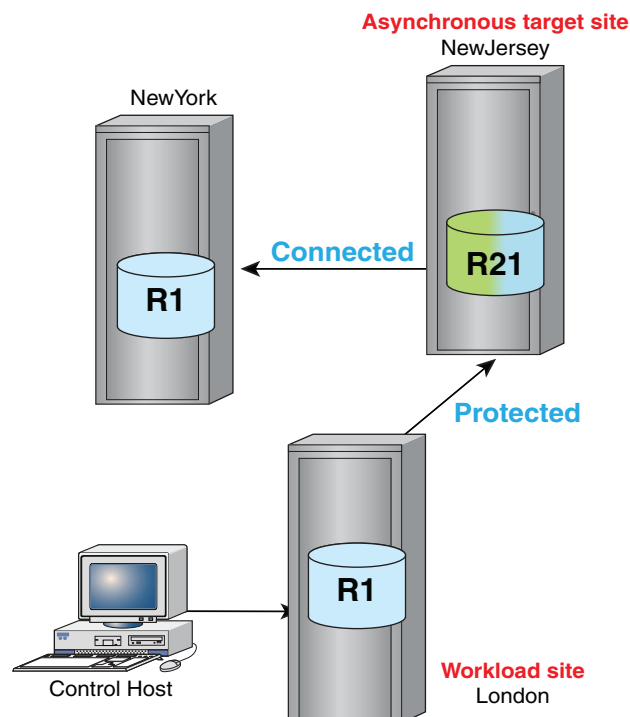
```
symstar -cg StarGrp connect -site NewJersey
symstar -cg StarGrp connect -site NewYork
```

5. Issue a `protect` command to protect one target site (NewJersey):

```
symstar -cg StarGrp protect -site NewJersey
```

The following image shows the resulting SRDF/Star configuration:

**Figure 92** Cascaded SRDF: after switch to asynchronous site, connect, and protect



If data is protected with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

[Step 7: \(Optional\) Add BCV devices to the SRDF/Star configuration on page 278](#)

[Switch back to the original workload site: concurrent SRDF/Star on page 316](#)

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):

#### Procedure

1. If London is in a Protected state, issue the `disconnect` command:

```
symstar -cg StarGrp disconnect -trip -site London
```

2. If the `disconnect` leaves London in a CleanReq state, issue the `cleanup` command:

```
symstar -cg StarGrp cleanup -site London
```

3. Issue 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.

---

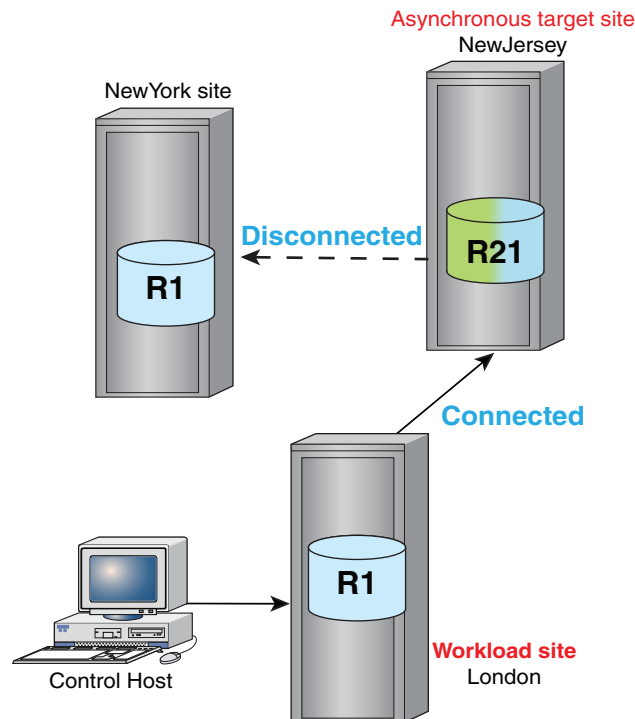
#### Note

The `connect` action is not required because the `switch` action specified that SRDF retrieve the remote data from the NewJersey site.

---

The following image shows the resulting SRDF/Star state:

**Figure 93** Cascaded SRDF: after switch to asynchronous site



If data is protected with BCV devices, make a TimeFinder/Clone or TimeFinder/Mirror copy.

See [Step 7: \(Optional\) Add BCV devices to the SRDF/Star configuration on page 278](#).

## Reconfiguration operations

This section describes the following topics:

- Reconfiguring from Cascaded SRDF/Star to Concurrent SRDF/Star
- Reconfiguring cascaded paths
- Reconfiguring from Concurrent SRDF/Star to Cascaded SRDF/Star
- Reconfiguring without halting the workload site

## Before you begin reconfiguration operations

- 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**, only the path to the asynchronous target site can be reconfigured.
- When the workload site is at **London**, the path to either the synchronous target site or the asynchronous target site can be reconfigured.
- If you do not want to halt the workload site, see [Reconfigure mode without halting the workload site on page 344](#).

## Reconfiguring mode: cascaded to concurrent

This section describes changing the SRDF/Star mode to concurrent from the synchronous or asynchronous workload site.

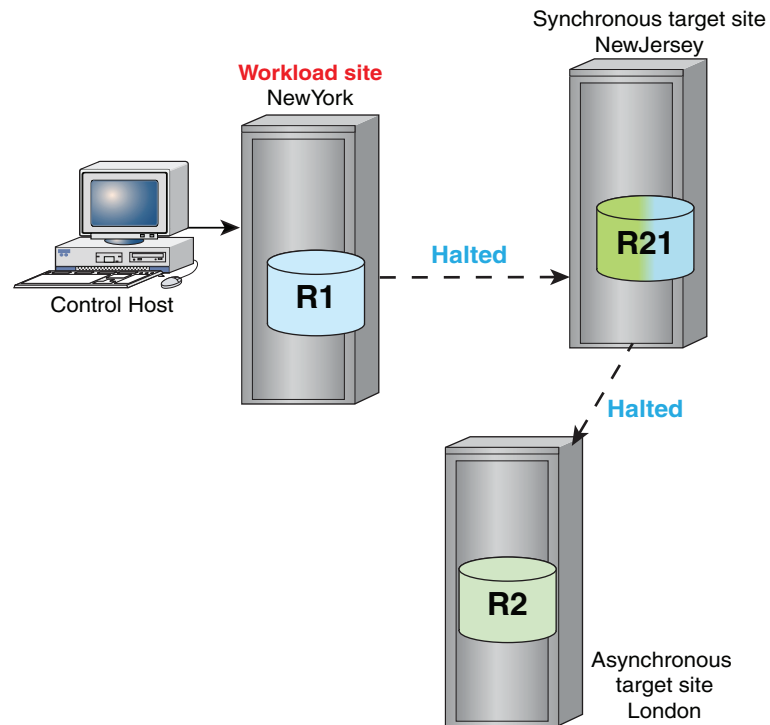
### Changing mode to concurrent: from synchronous workload site

#### Procedure

1. From a control host at the workload site, issue the `halt` command to stop SRDF:

```
symstar -cg StarGrp halt
```

The following image shows the resulting SRDF/Star state:

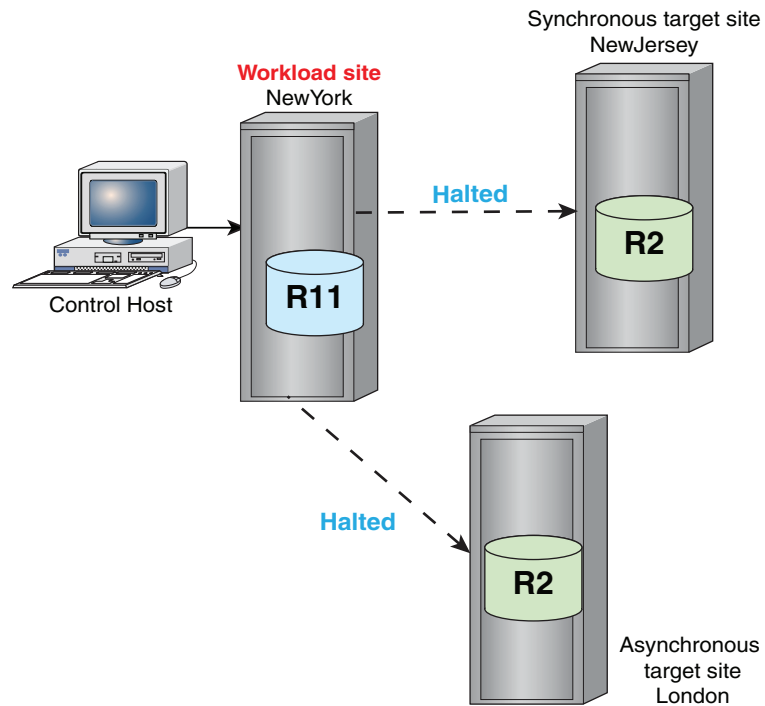
**Figure 94** Halted cascaded SRDF/Star

2. Issue the `symstar reconfigure` command to reconfigure the NewYork -> NewJersey -> London path to NewYork -> London:

```
symstar -cg StarGrp reconfigure -site London -path NewYork:London
```

See [Recover using reconfigure operations on page 300](#).

The following image shows the resulting SRDF/Star state:

**Figure 95** After reconfiguration to concurrent

## Changing mode to concurrent: from asynchronous workload site

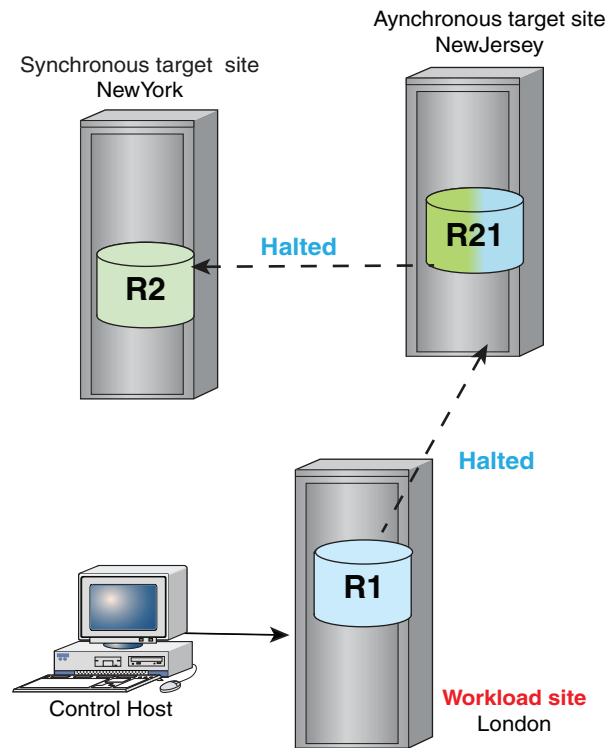
### Procedure

1. From a control host at the workload site, issue the `halt` command to stop SRDF:

```
symstar -cg StarGrp halt
```

The following image shows the resulting SRDF/Star state:



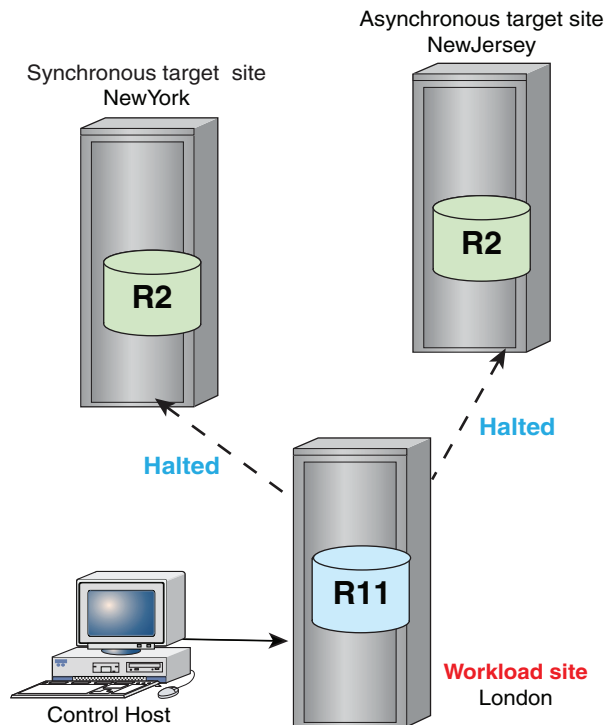
**Figure 96** Halted cascaded SRDF/Star

2. Issue the `symstar reconfigure` command to reconfigure the London -> NewJersey -> NewYork path to London -> NewYork:

```
symstar -cg StarGrp reconfigure -site NewYork -path London:NewYork
```

See [Recover using reconfigure operations on page 300](#).

The following image shows the resulting SRDF/Star state:

**Figure 97** 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.

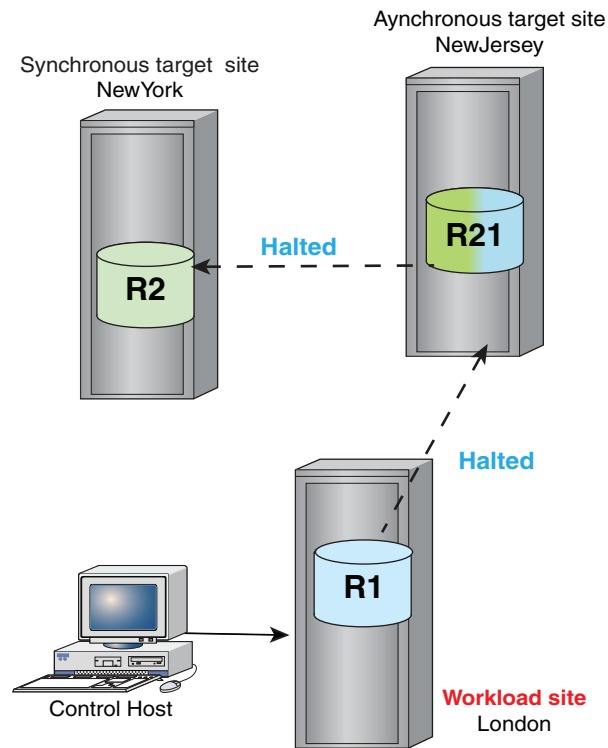
Complete the following steps to reconfigure the path to the synchronous target site (NewJersey) when the workload site is at London.

### Procedure

1. From a control host at the workload site, issue the `halt` command to stop SRDF:

```
symstar -cg StarGrp halt
```

The following image shows the resulting SRDF/Star state:

**Figure 98** Halted cascaded SRDF/Star

2. Issue the `symstar reconfigure` command with `-path` and `-remove` options to reconfigure the path from:

London → NewJersey → NewYork

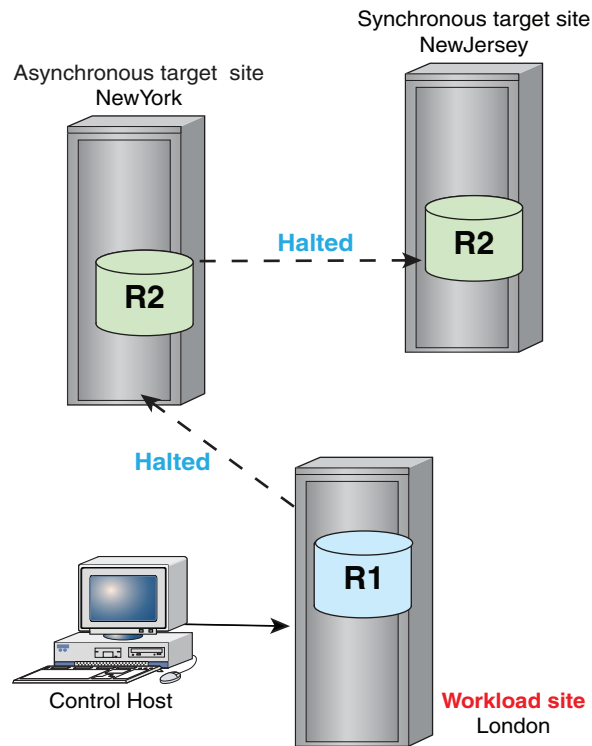
to:

London → NewYork → NewJersey:

```
symstar -cg StarGrp reconfigure -site NewYork -path London:NewYork
-remove London:NewJersey
```

See [Recover using reconfigure operations on page 300](#).

The following image shows the resulting SRDF/Star state:

**Figure 99** After cascaded path reconfiguration

## Reconfiguring mode: concurrent to cascaded

This section describes changing the SRDF/Star mode to cascaded from the synchronous or asynchronous workload site.

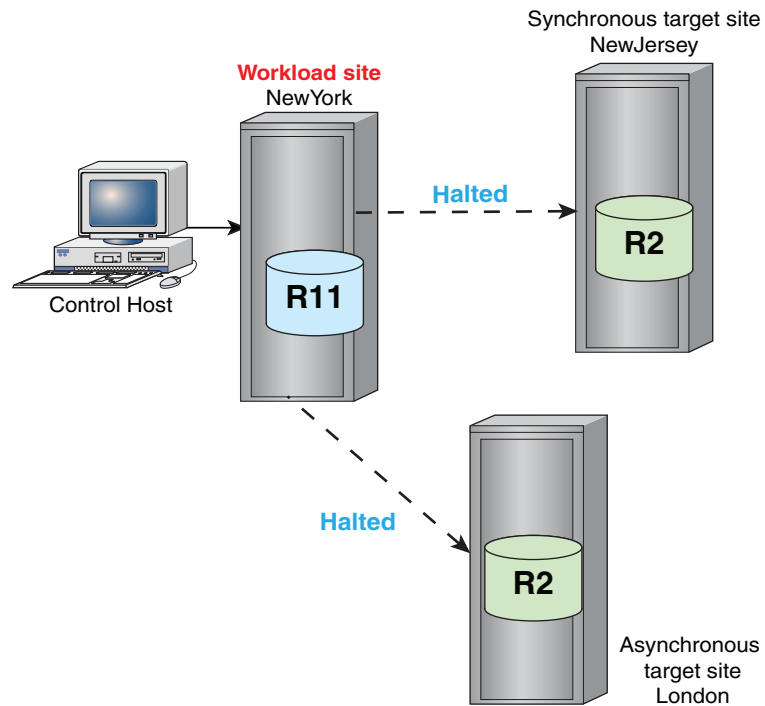
### Changing mode to cascaded: from synchronous workload site

#### Procedure

1. From a control host at the workload site, issue the `halt` command to stop SRDF:

```
symstar -cg StarGrp halt
```

The following image shows the resulting SRDF/Star state:

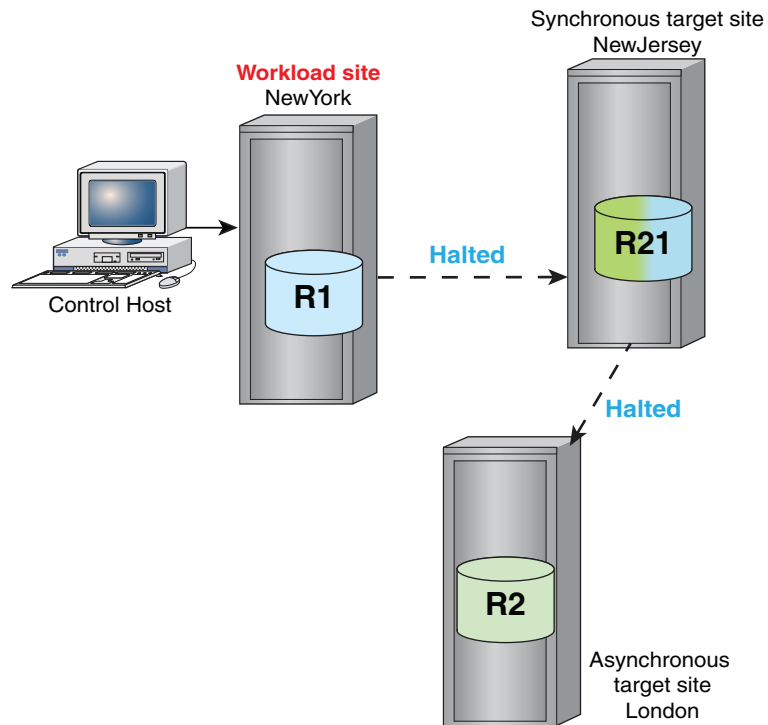
**Figure 100** Halted concurrent SRDF/Star

2. Issue 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 [Recover using reconfigure operations on page 300](#).

The following image shows the resulting SRDF/Star state:

**Figure 101** After reconfiguration to cascaded

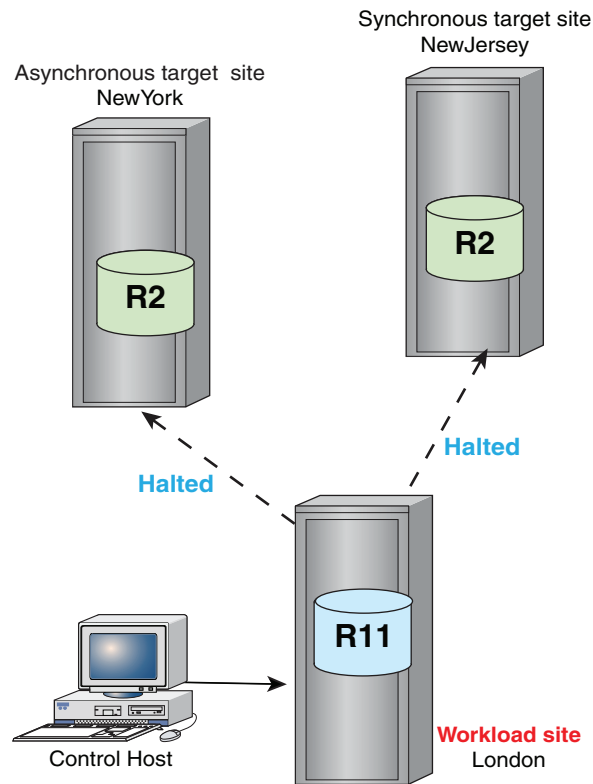
### Changing mode to cascaded: from asynchronous workload site

#### Procedure

1. From a control host at the workload site, issue the `halt` command to stop SRDF:

```
symstar -cg StarGrp halt
```

The following image shows the resulting SRDF/Star state:

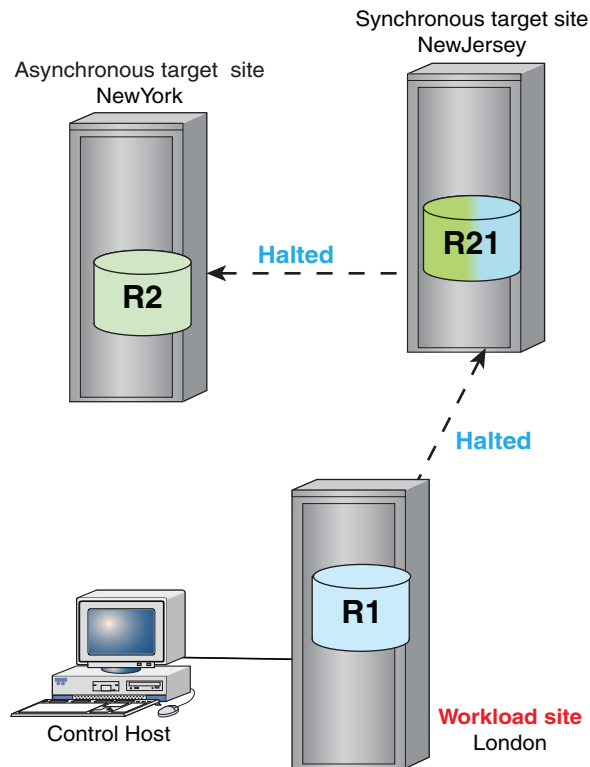
**Figure 102** Halted concurrent SRDF/Star

2. Issue the `symstar reconfigure` command to reconfigure the concurrent path from London → New York to cascaded path London → New Jersey → New York:

```
symstar -cg StarGrp reconfigure -site London -path NewJersey:London
```

See [Recover using reconfigure operations on page 300](#).

The following image shows the resulting SRDF/Star state:

**Figure 103** After reconfiguration to cascaded

## Reconfigure mode without halting the workload site

This section describes the following topics:

- Reconfiguring cascaded mode to concurrent mode
- Reconfiguring concurrent mode to cascaded mode

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 320](#).

### NOTICE

These operations take the system out of the STAR Protected state.

Once reconfiguration is complete, re-enable STAR protection.

## Reconfigure cascaded mode to concurrent mode

In the following example:

- The SRDF/Star environment is operating in 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`.

---

## Reconfigure 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 SRDF/Star configuration with R22 devices
- Transitioning SRDF/Star to use R22 devices

### Before you begin SRDF/Star configuration with R22 devices

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 274](#).

- 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 (synchronous 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 synchronous 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, and so on. 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 synchronous target,
- workload to asynchronous target, or
- workload to synchronous target,
- synchronous 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
```

## Transition 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.

Issue 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 synchronous 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

```

Issue 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:

```

Issue the `symstar query` command to verify that adding recovery SRDF pairs was the last action performed. For example (truncated output):

```

symstar -cg CgName query

...
Last Action Performed      :ConfigureAddRcvryRDFPair
Last Action Status         :Successful
Last Action timestamp      :03/15/2008_12:29:37

```



# CHAPTER 10

## Device Migration Operations

This chapter describes the following topics:

|                                                                  |     |
|------------------------------------------------------------------|-----|
| • <a href="#">Device Migration operations overview</a> .....     | 350 |
| • <a href="#">Device Migration operations requirements</a> ..... | 350 |
| • <a href="#">R1 device migration</a> .....                      | 351 |
| • <a href="#">R2 device migration</a> .....                      | 353 |
| • <a href="#">R1 and R2 migration procedures</a> .....           | 357 |
| • <a href="#">SRDF pair states for migration</a> .....           | 368 |

## Device Migration operations 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.

## Device Migration operations 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

- Devices that are part of an SRDF/Metro configuration cannot be migrated.
- 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.

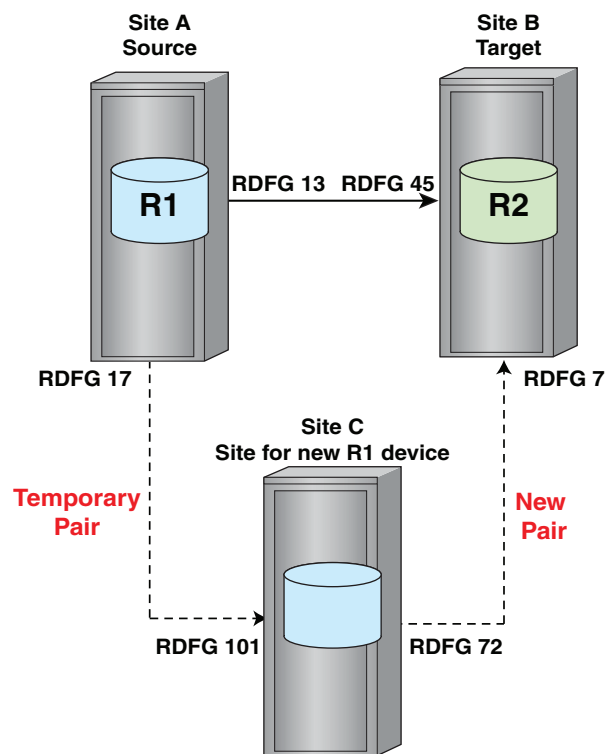
This section describes the steps to complete an R1 migration, including:

- [Configure a temporary SRDF group on page 351](#) and R1 device to enable the migration.
- [Establish a concurrent SRDF relationship on page 355](#) to transfer data to the from the old R1 device to the device that will become the new R1.
- [Replacing the R1 device on page 352](#) 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 104** R1 migration: configuration setup



In the preceding example:

- 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.

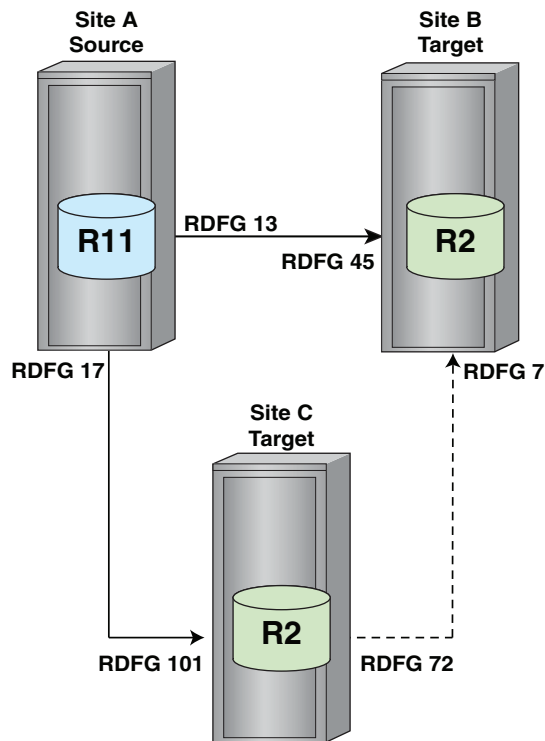
- A temporary SRDF group (RDFG 17) is used to synchronize data from the existing R1 to the new device in Site C.

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 105** R1 migration: establishing a concurrent relationship



In the preceding example:

- The R1 device becomes the concurrent R11 device writing to two R2 devices.
- 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.

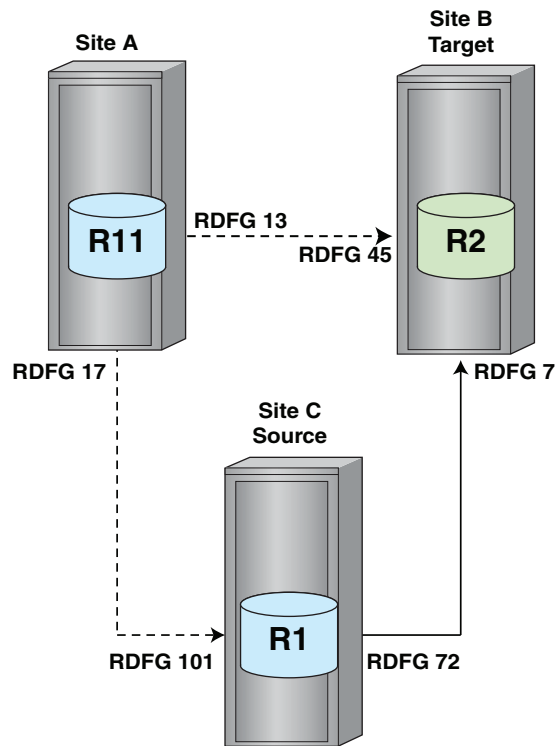
---

## Replacing the R1 device

### Procedure

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 106** R1 migration: replacing the source device

The `symrdf migrate -replace R1` command executes the following actions:

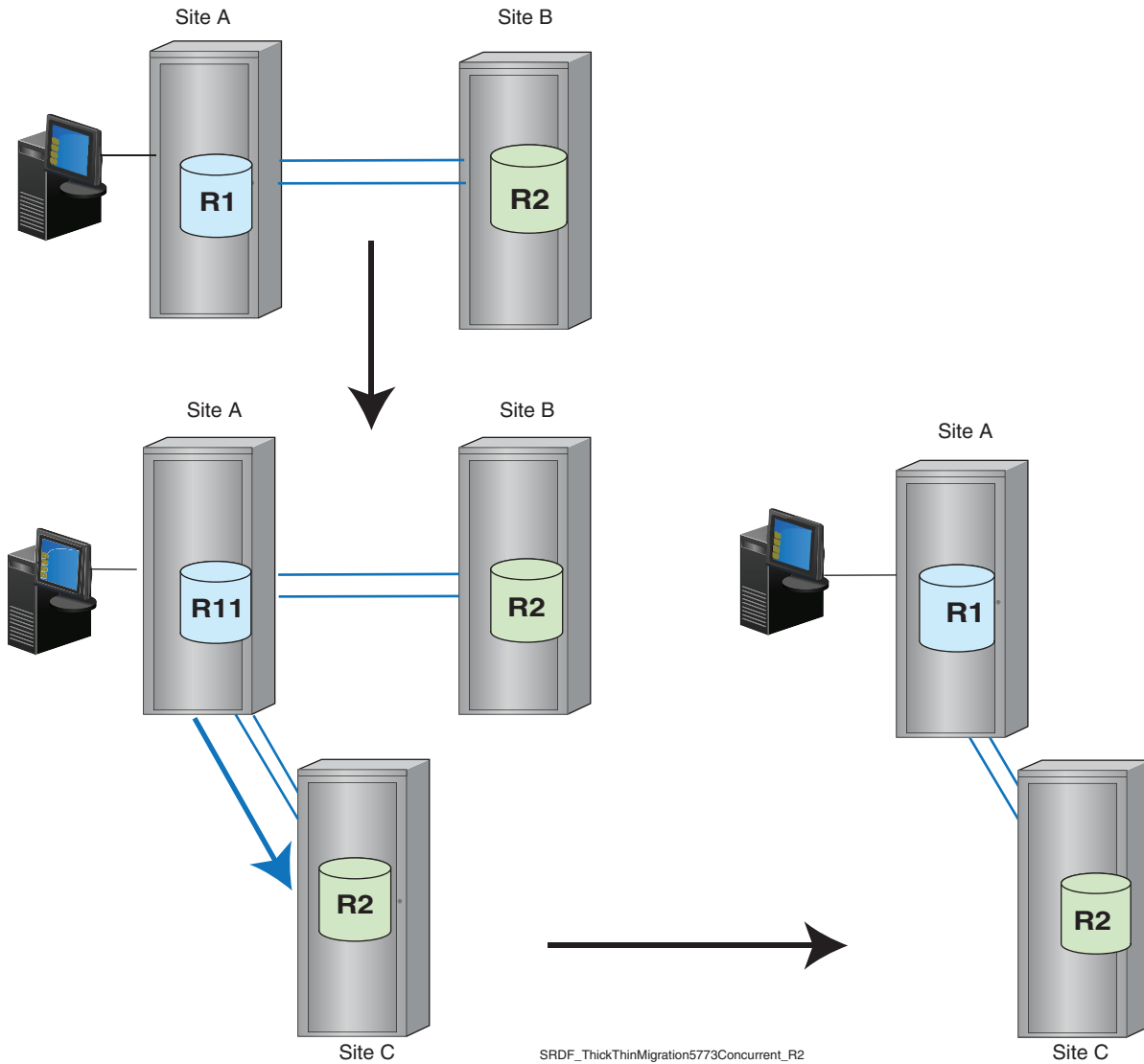
- Sets the source device to USR-NR (user not ready).  
This prevents applications writing to or reading from the R1 device.
- Verifies the devices are in the correct pair state for replacement.  
See also [SRDF pair states for migration on page 368](#).
- (If applicable) Waits until all invalid tracks are cleared.
- (If applicable) Drains the SRDF/A session.
- Removes the SRDF pairing between the devices on the current R11 (Site A) and the original R2 (Site B).
- Removes the SRDF pairing between the devices on the current R11 (Site A) and the new R2 (Site C).
- 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.  
No additional copying of data is required between this SRDF pair because data is already the same on both devices.
- 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.

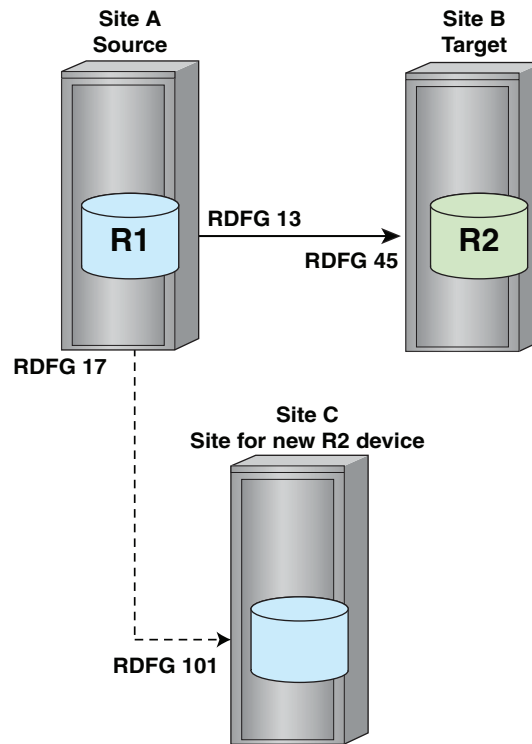
**Figure 107** Migrating R2 devices

This section describes the steps to complete an R2 migration, including:

- [Configure setup for R2 migration on page 354](#)
- [Establish a concurrent SRDF relationship on page 355](#) to transfer data to the from the R1 device to the device that will become the new R2.
- [Replacing the R2 device on page 356](#) with the newly-populated device in the SRDF pair.

## Configure setup for R2 migration

Configure a replacement R2 as a non-SRDF device:

**Figure 108** R2 migration: configuration setup

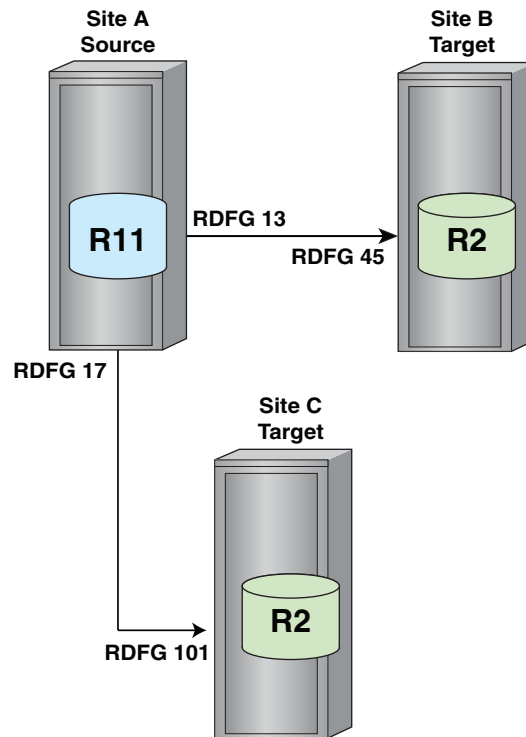
In the preceding example:

- 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 109** 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 the preceding example, 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.

---

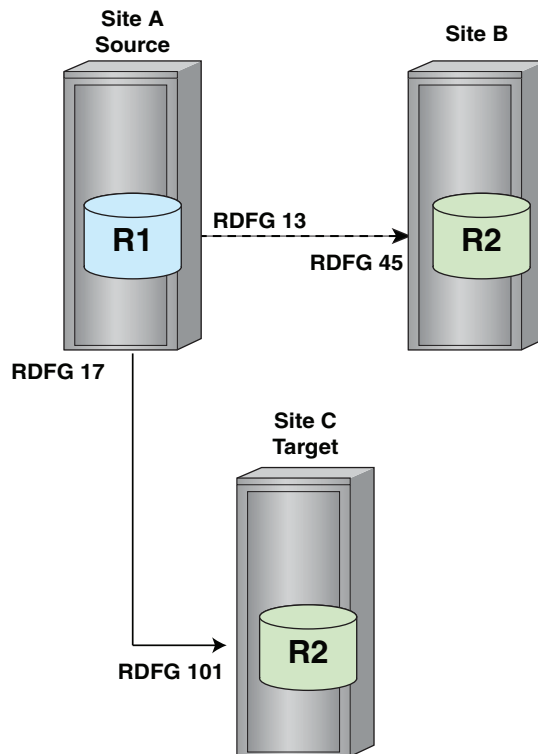
#### Note

It may be necessary to modify existing device group or composite group scripts to accommodate the new configuration.

---

## Replacing 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 110** 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 368](#) 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 R1 and R2 migration

- 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 368.

## Restrictions for R1 and R2 migration

### 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.

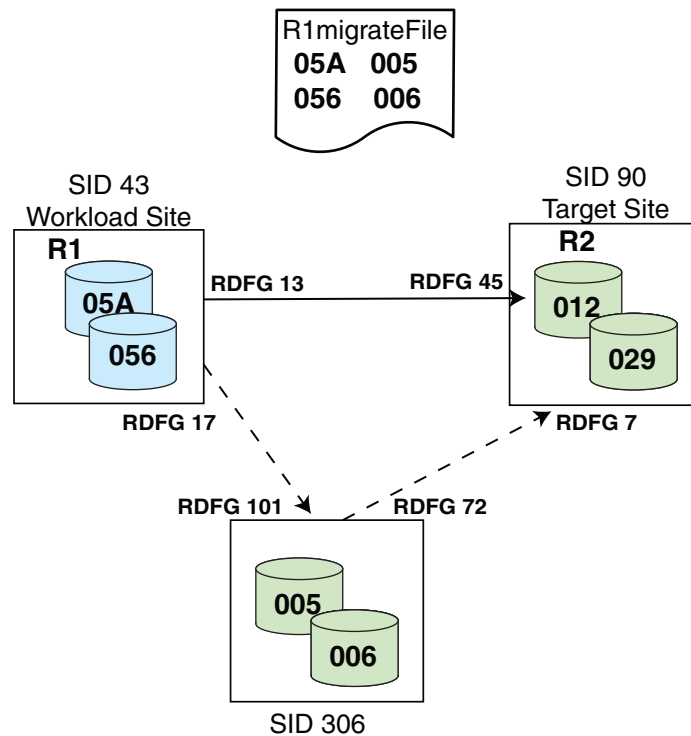
[Sample procedure: migrating R1 devices](#) on page 359, explains the procedure for an SRDF consistency protection group enabled at the composite-group level.

## 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 111** R1 migration example: Initial configuration



The preceding image 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    |            |
|------------------|-------|----|--------|--------|------------------|----|--------|--------|----------|------------|
| -----            |       |    |        |        | -----            |    |        |        | -----    |            |
| Standard         | ST    |    |        |        | LI               | ST |        |        |          |            |
|                  | 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        |            |
| MBs              | 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.

[Create a device file](#) on [page 101](#) 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

It may be necessary 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

The `-force` option is used when 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...

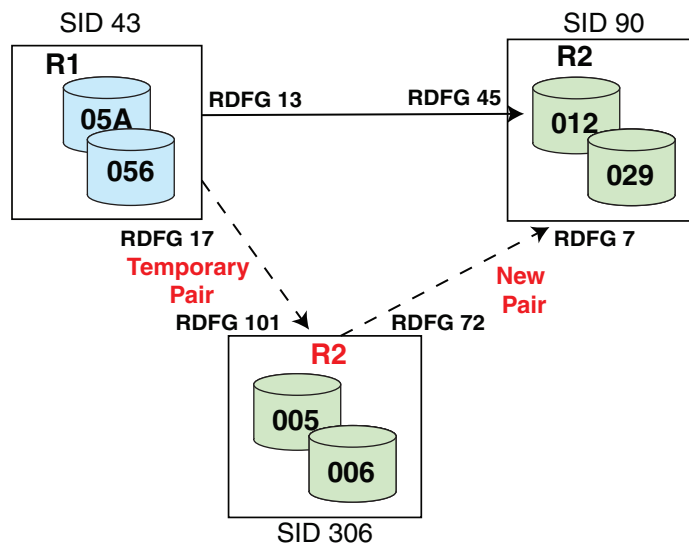
```
Migrate Setup for R1 device(s) in (043,017).....Started.
Create RDF Pair in (0043,017).....Started.
Create RDF Pair in (0043,017).....Done.
Mark target device(s) in (0043,017) for full copy from source....Started.
Devices: 06F0-06FF in (0043,017).....Marked.
Mark target device(s) in (0043, 017) for 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 112** Concurrent SRDF relationship



In the preceding image:

- 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 `symcgc disable` command to disable it.

To disable SRDF consistency protection for composite group MigrateRDF:

```
symcrg -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 Symmetrix 90:

```
symrdf -sid 043 -rdfg 17 -f RlmigrateFile migrate -replace r1 -config pair -new_rdfg 72
```

An RDF 'Migrate Replace R1' operation execution is in progress for device file 'RlmigrateFile'. 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 'RlmigrateFile'.

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 MigrateRDF enable
```

A consistency 'Enable' operation execution is in progress for composite group 'MigrateRDF'. Please wait...

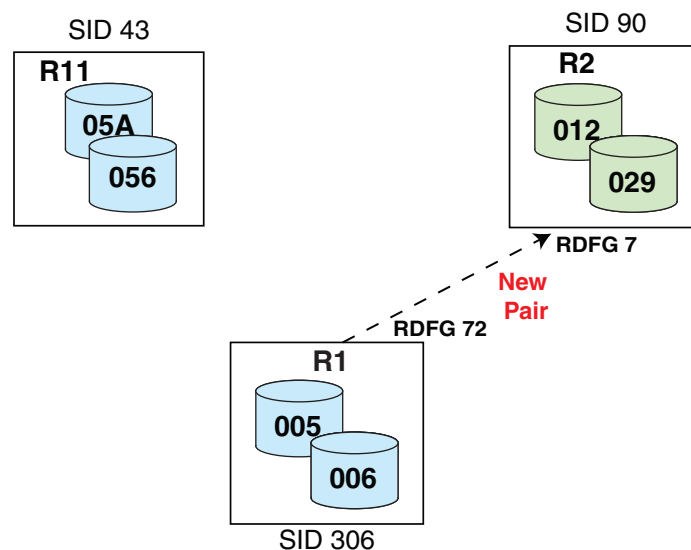
The consistency 'Enable' operation successfully executed for composite group 'MigrateRDF'.

When migration is complete (as shown in the following image):

- 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 113** 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
}
```

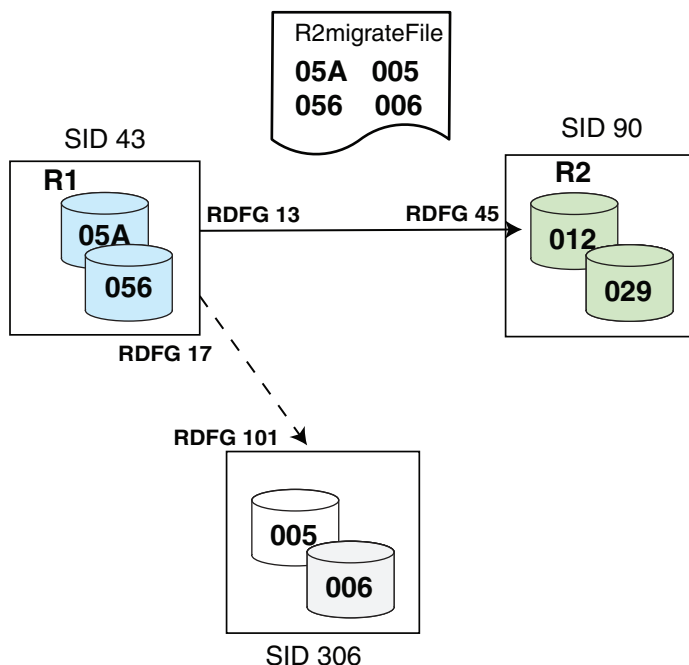
```
Symmetrix ID                : 000190100306 (Microcode Version: 5773)
Remote Symmetrix ID        : 000192600090 (Microcode Version: 5874)
RDF (RA) Group Number       : 3 (02) - siteb
RDFA Info:
```

```
{
    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  |                  |
|------------------|-------|----|--------|--------|------------------|-------|--------|--------|--------|------------------|
| -----            |       |    |        |        | -----            |       |        |        | -----  |                  |
| Standard         | Sym   | ST | R1 Inv | R2 Inv | LI               | ST    | R1 Inv | R2 Inv |        |                  |
| Logical          | Dev   | A  | Tracks | Tracks | N                | A     | Tracks | Tracks |        |                  |
| Device           | Dev   | E  | Tracks | Tracks | S                | Dev   | E      | Tracks | Tracks | RDF Pair         |
| -----            |       |    |        |        | -----            |       |        |        | -----  |                  |
| 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 114** R2 migration example: Initial configuration

The preceding example 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.

[Create a device file on page 101](#) 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.

---

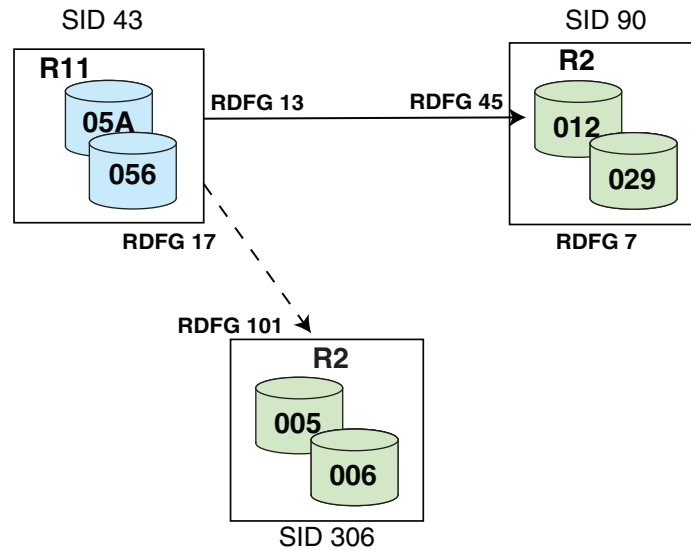
#### Note

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 115** Concurrent SRDF relationship

In the preceding example:

- 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

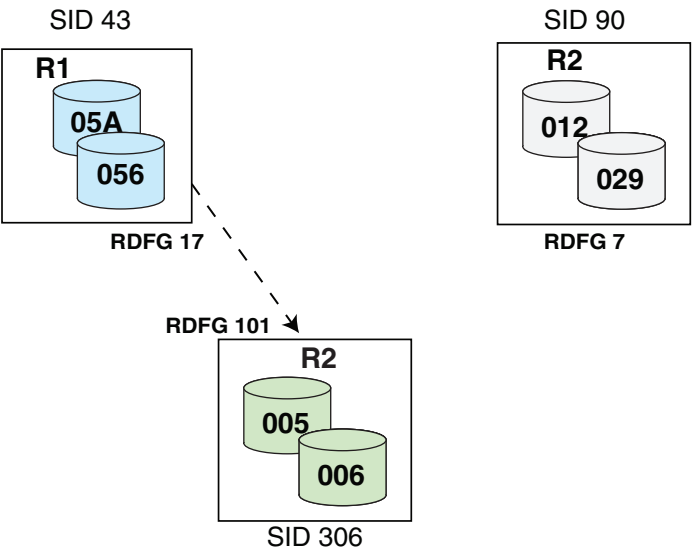
After replacing R2, you must modify device groups and/or composite groups to remove all BCVs, VDEVs, TGTs from the original R2 and then add appropriate counterparts to the new R2. You must also recreate any TF/Mirror, TF/Snap, TF/Clone, Open Replicator, and Federated Live Migration sessions on the new 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 -rdfig 17 migrate -replace R2 -config pair
```

**Figure 116** Migrated R2 devices



When migration is complete, the array 306 devices become the R2 devices and are paired with the R1 devices in Symmetrix 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 be in a specific SRDF state to perform certain migration control operations.

The following table lists the applicable pair states for `symrdf migrate -setup` for an R1 and an R2 migration.

**Table 39** SRDF migrate -setup control operation and applicable pair states

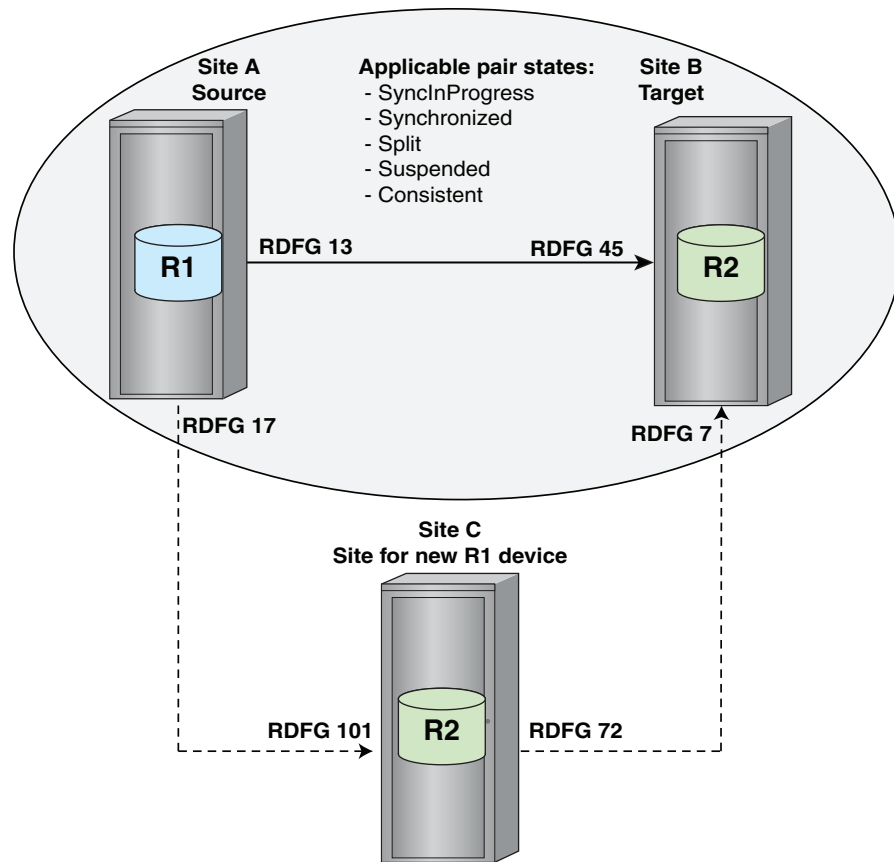
| Control operation: | Pair state: existing R1->R2 |              |                |                |             |                           |                           |            |              |         |            |
|--------------------|-----------------------------|--------------|----------------|----------------|-------------|---------------------------|---------------------------|------------|--------------|---------|------------|
|                    | SyncInProg                  | Synchronized | Split          | Suspended      | Failed over | Partitioned1 <sup>a</sup> | Partitioned2 <sup>b</sup> | R1 updated | R1 updinprog | Invalid | Consistent |
| migrate -setup     | P                           | P            | P <sup>c</sup> | P <sup>c</sup> |             |                           |                           |            |              |         | P          |

- a. The remote array is in the SYMAPI database (it was discovered).
- b. The remote array is not in the SYMAPI database (it was not discovered or was removed).
- c. Only when replacing the R2 devices.

## Pair states for migrate -setup

The following image shows a sample configuration for an R1 migration:

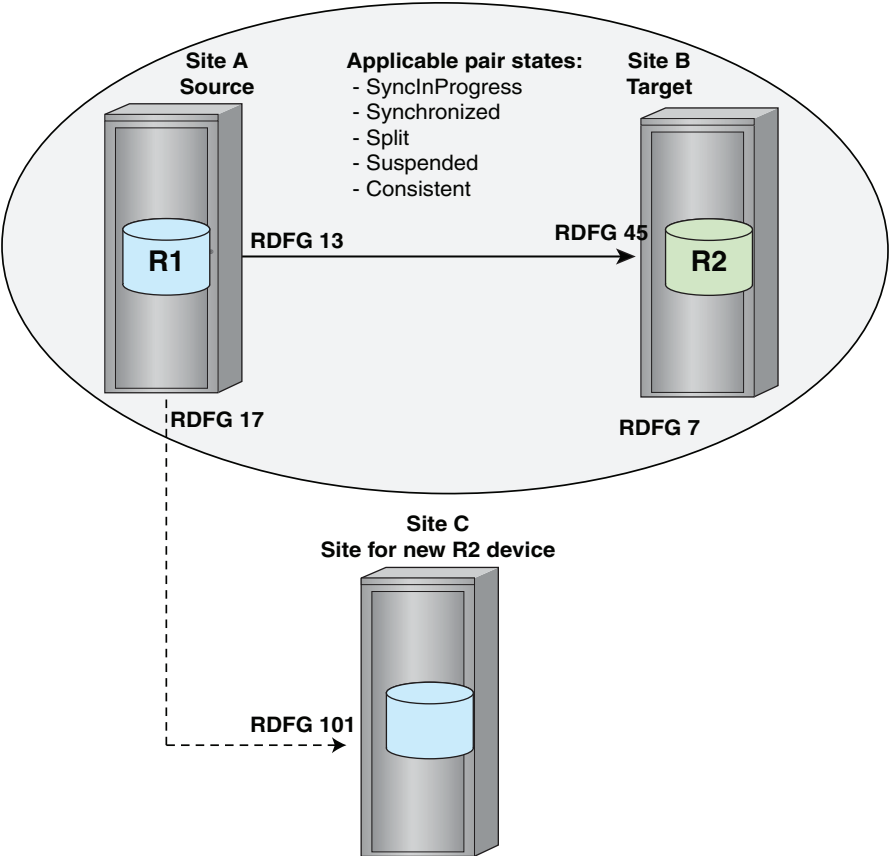


**Figure 117** R1 migration: applicable R1/R2 pair states for migrate -setup

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.

The following image shows a sample configuration for an R2 migration:

**Figure 118** 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 119 on page 371](#) shows the SRDF pair state required before replacing an R1, the R11 and its existing device.

[Figure 120 on page 372](#) 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.

The following table lists the applicable pair states for `symrdf migrate -replace` for an R1 and an R2 migration.

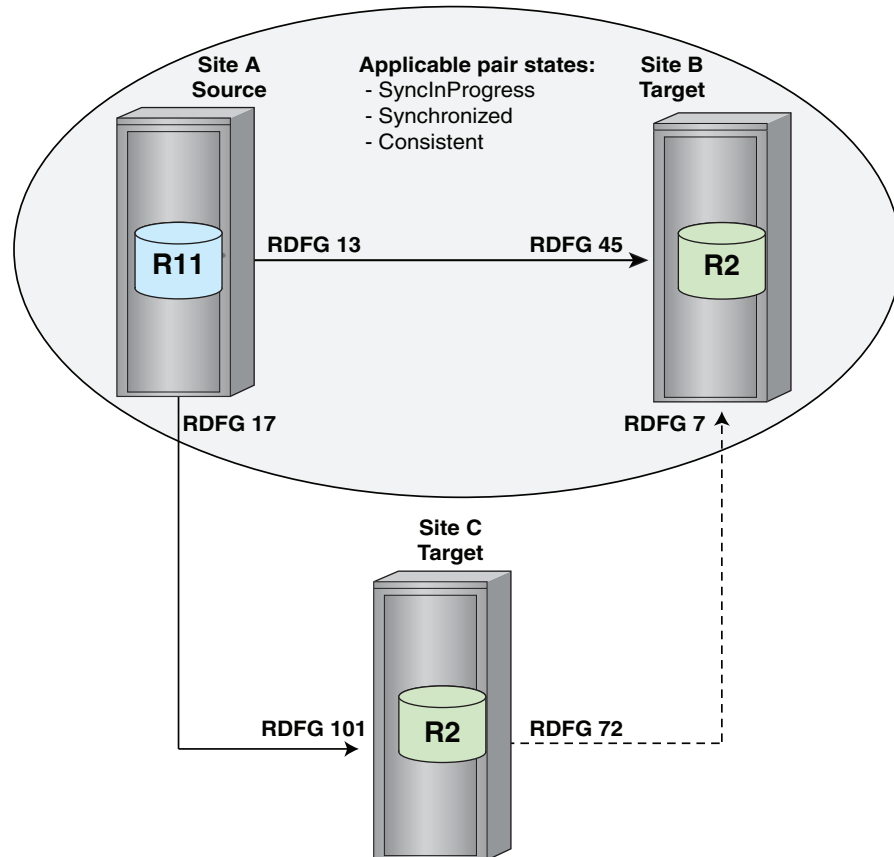
**Table 40** SRDF migrate -replace control operation and applicable pair states

| Control operation: | Pair state: Existing ->R2 |              |       |           |             |                           |                           |            |              |         |            |
|--------------------|---------------------------|--------------|-------|-----------|-------------|---------------------------|---------------------------|------------|--------------|---------|------------|
|                    | SyncInProg                | Synchronized | Split | Suspended | Failed over | Partitioned1 <sub>a</sub> | Partitioned2 <sub>b</sub> | R1 updated | R1 updinprog | Invalid | Consistent |
| migrate -replace   | P                         | P            | P     | P         |             |                           |                           |            |              |         | P          |

**Table 40** SRDF migrate -replace control operation and applicable pair states

- a. The remote array is in the SYMAPI database (it was discovered).
- b. The remote array is not in the SYMAPI database (it was not discovered or was removed).

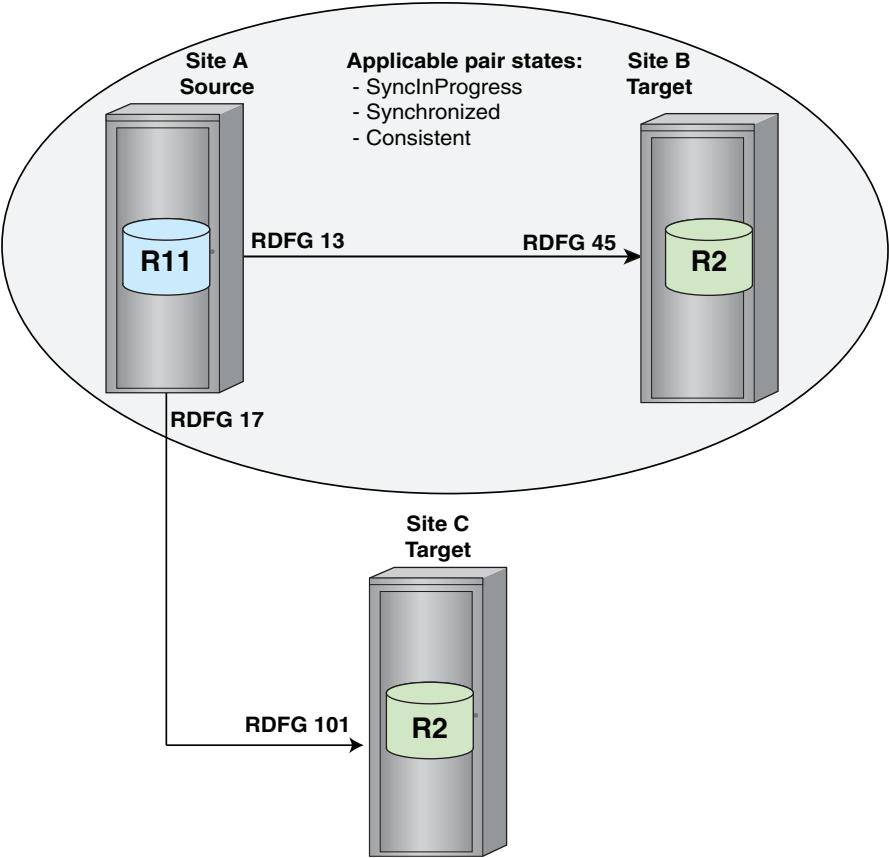
The following image shows a sample concurrent SRDF configuration for an R1 migration:.

**Figure 119** 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.

The following image shows a sample concurrent SRDF configuration for an R2 migration:

**Figure 120** 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 121 on page 373](#). 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 122 on page 374](#).

The following table lists the applicable pair states for `symrdf migrate -replace` for an R1 and an R2 migration.

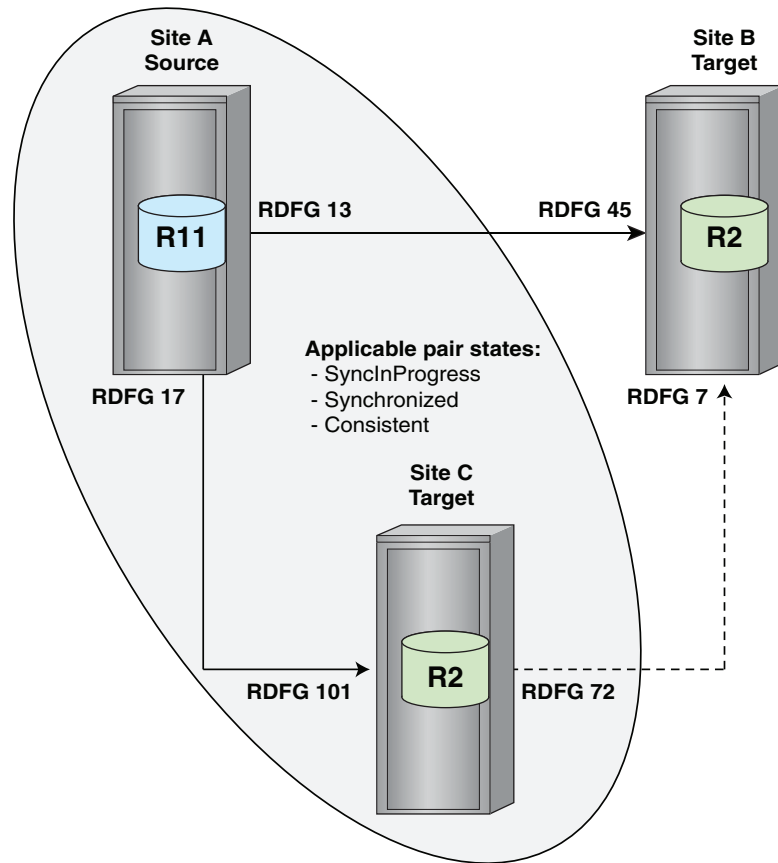
**Table 41** SRDF migrate -replace control operation and applicable pair states

|                    | Pair state: Temporary or New ->R2 |              |       |           |             |                           |                           |            |              |         |            |
|--------------------|-----------------------------------|--------------|-------|-----------|-------------|---------------------------|---------------------------|------------|--------------|---------|------------|
|                    | SyncInProg                        | Synchronized | Split | Suspended | Failed over | Partitioned1 <sup>a</sup> | Partitioned2 <sup>b</sup> | R1 updated | R1 updinprog | Invalid | Consistent |
| Control operation: |                                   |              |       |           |             |                           |                           |            |              |         |            |
| migrate -replace   | P                                 | P            |       |           |             |                           |                           |            |              |         | P          |

a. The remote array is in the SYMAPI database (it was discovered).  
b. The remote array is not in the SYMAPI database (it was not discovered or was removed).

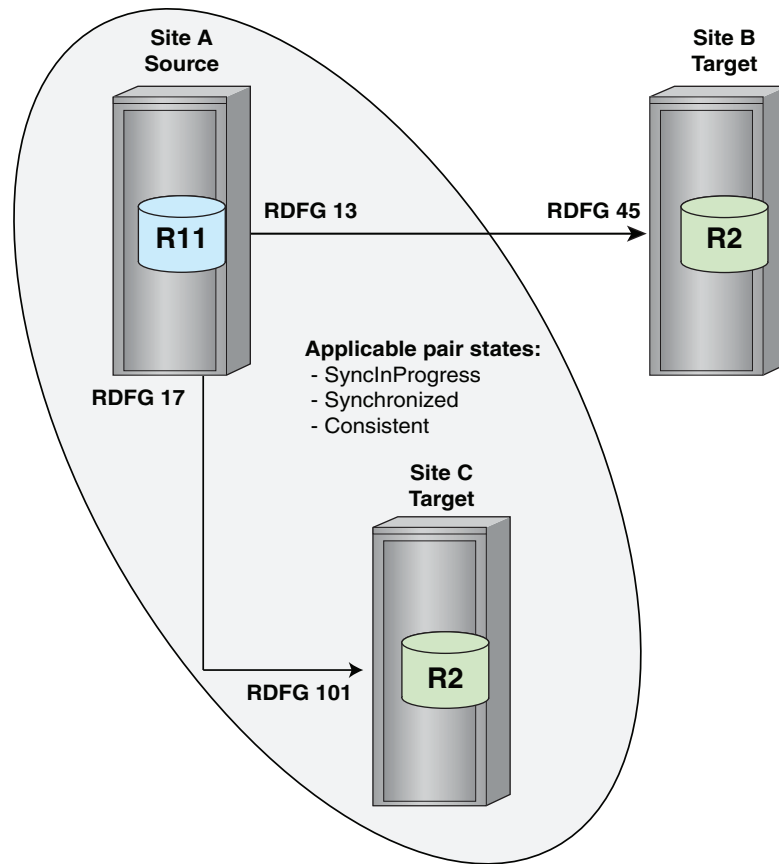
The following image shows a sample concurrent SRDF configuration for an R1 migration.

**Figure 121** 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.

The following image shows a sample concurrent SRDF configuration for an R2 migration:

**Figure 122** 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 11

## SRDF/Automated Replication

This chapter describes the following topics:

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| • <a href="#">SRDF/Automated Replication overview</a> .....             | 376 |
| • <a href="#">SRDF/Automated Replication operations</a> .....           | 377 |
| • <a href="#">Clustered SRDF/AR</a> .....                               | 386 |
| • <a href="#">Set symreplicate parameters in the options file</a> ..... | 389 |
| • <a href="#">Manage locked devices</a> .....                           | 394 |

## SRDF/Automated Replication 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

Multi-hop SRDF/AR requires Engenuity 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/Automated Replication

- 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. [Manage locked devices on page 394](#) provides more information.

---

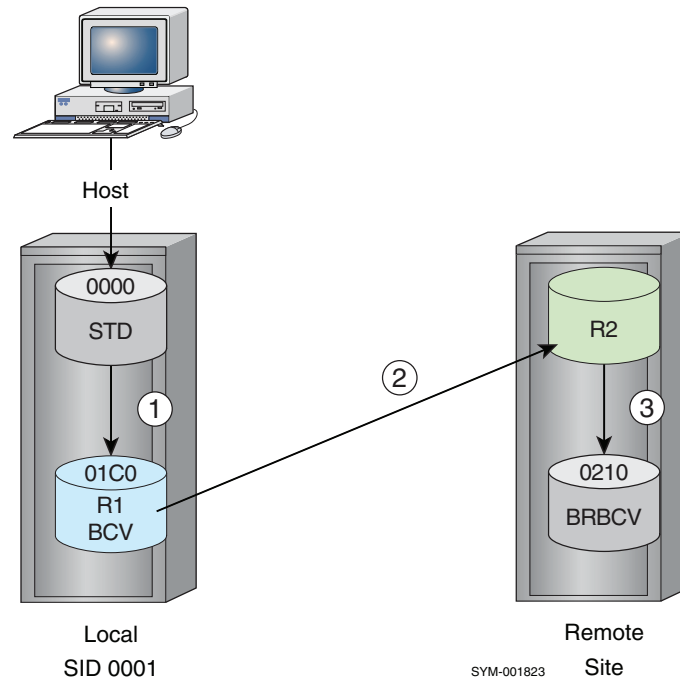


# SRDF/Automated Replication operations

## Configure single-hop sessions

The following image shows how **symreplicate** copies data in a single-hop configuration for a complete copy cycle:

**Figure 123** 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.

[Setting the symreplicate control parameters on page 390](#) provides 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:

## Procedure

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:

- [Setting up single hop manually on page 380](#)
- [Setting up multi-hop manually on page 382](#)
- [Setting the symreplicate control parameters on page 390](#)

#### 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
```

#### -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.

#### -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.

---

#### Note

Single-hop replication does a full optimization on all RA groups.

---

#### Syntax

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).

### Examples

```
symreplicate setup -g DgName -optimize
```

```
symreplicate setup -cg CgName -optimize_rag
```

### symreplicate 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.

[Setting the symreplicate control parameters on page 390](#) provides more information.

## Setting up single hop manually

To manually reproduce the single-hop replication cycle using a sequence of SRDF and TimeFinder CLI commands:

### Procedure

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

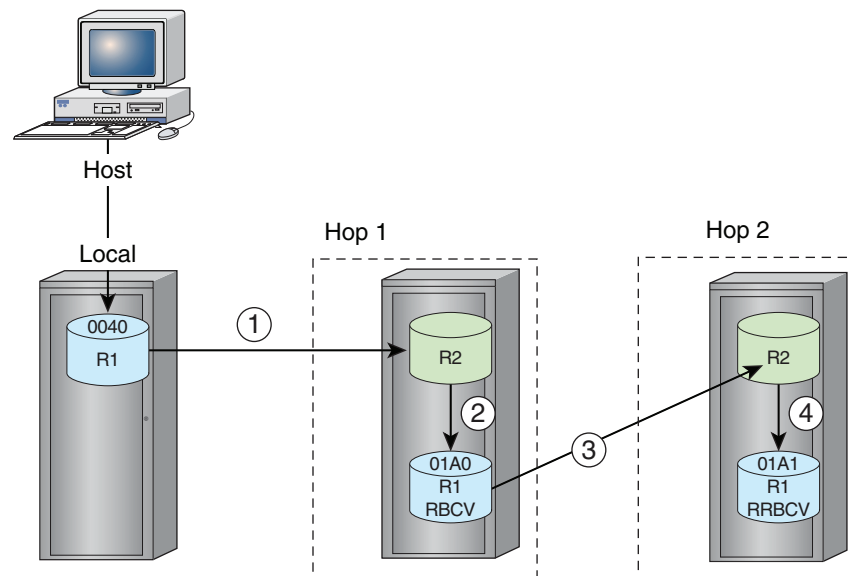
You may have to include additional command options in some of the above steps (for example, `establish -full` for BCV pairs without relationships).

---

## Configure multi-hop sessions

The following image shows a complete **symreplicate** copy cycle in a multi-hop configuration:

**Figure 124** Automated data copy path in multi-hop SRDF



Data copy paths in the image above 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 RRBCV

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
```

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 390](#) provides more information.

## Setting up for a multi-hop configuration

To set up a multi-hop `symreplicate` session:

### Procedure

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 378](#) provides more information.

## Setting up multi-hop manually

To manually reproduce the multi-hop replication cycle using a sequence of SRDF and TimeFinder CLI commands:

## Procedure

1. Wait for any ongoing establish to complete.
2. Split the BCV pairs (2b in [Figure 124 on page 381](#)):

```
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 124 on page 381](#)), and establish the BCV pairs (2b in [Figure 124 on page 381](#)):

```
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 124 on page 381](#)):

```
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.

Optionally, 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).

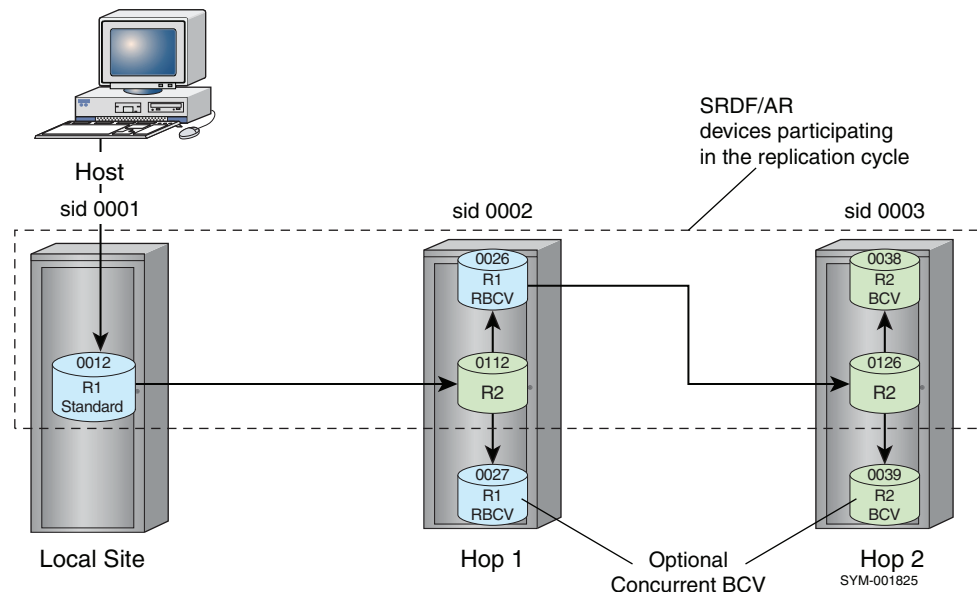
---

## 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 125** Concurrent BCV in a multi-hop configuration



In the image above, 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:

### Parameters

`SYMCLI_REPLICATE_CYCLE=CycleTime`

*CycleTime* is a timer that specifies 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* specifies the number of replication cycles (copies) to perform before `symreplicate` exits. For example, a value of zero (the default) results in continuous cycling until the `symreplicate stop` command is issued.

`SYMCLI_REPLICATE_CYCLE_DELAY= Delay`



*Delay* specifies 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* specifies the behavior when the actual copy time of data and/or data transfer is so large that it exceeds the *CycleTime* value. 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.)

#### Example

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`).

## Set 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.

The following table lists two parameter setups for an initial `symreplicate` session trial:

**Table 42** Initial setups for cycle timing parameters

|                                                                                                                                            |                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>SYMCLI_REPLICATE_CYCLE=60</code><br><code>SYMCLI_REPLICATE_CYCLE_DELAY=0</code><br><code>SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT</code> | Every hour if possible, or every 2, or 3 hours based on data throughput and size.                     |
| <code>SYMCLI_REPLICATE_CYCLE=0</code><br><code>SYMCLI_REPLICATE_CYCLE_DELAY=60</code>                                                      | Cycle through the first copy, then wait 60 minutes (delay), and then another cycle, delay, and so on. |

## View cycle time and invalid track statistics

#### Syntax

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)

## Options

### -log

Write information to a specified log file.

### -cycle

Display only cycle time statistics for the last SRDF/AR cycle time, the maximum cycle time and the average cycle time.

### -itrks

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.

### -all

(default) 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)
```

## Log 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.

[Setting the symreplicate control parameters on page 390](#) 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.

## Write log files to a specified SFS

### Syntax

Use the `symreplicate start` command with the `-sid` and `-log` options to write the log file to the SFS. The following options must be specified:

### Options

#### **-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.

### Restrictions

- If Symmetrix ID (`-sid`) is not specified 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 a user log file name (`-log LogFilename`) is specified when a session is started, the `-log` option must be specified for all other commands in the session sequence.
- If only the group name (`-g`, `-cg`) is specified when a session is started:
  - The log file is given the same name as the group,
  - Specify only the `-g` or `-cg` option for all other commands in the session sequence.

### HYPERMAX OS restrictions

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`

### Example

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
```

## Restart 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.

### Syntax

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.

You do not need to specify the device or composite group name (`-g`, `-cg`) on the host where the session is restarted.

**Options****-recover**

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.

**Example**

To restart the SRDF/AR session from another local host:

```
symreplicate restart -g session1 -log srdfar1.log -sid 201 -recover
```

**List log files written to the SFS****Syntax**

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
```

**Show log files written to SFS****Syntax**

Use the `symreplicate show -log LogfileName -sid SID -all` command to display the information content of a particular log file.

EMC Solutions Enabler CLI Command Reference provides more information.

**Options****-log**

Required. Log filename.

**-sid**

Required. Symmetrix ID.

**-args**

Display only command line arguments.

**-devs**

Display only devices.

**-opts**

Display only options.

**-all**

(default) Display all available information contained in the log.

**Example**

To display the log file `srdfar1.log` at SID 201:

```
symreplicate show -log srdfar1.log -sid 201 -all
```

## Delete a log file written to SFS

### Syntax

Use the `symreplicate delete -log LogFile.log` command to delete the specified log file written to SFS.

Specify either the group name (`-g`, `-cg`) or the log filename (`-log`) depending on whether a user log name was specified when the session was started.

### Example

To delete log file `srdfar1.log` written to the SFS:

```
symreplicate delete -log srdfar1.log
```

## Set symreplicate parameters in the options file

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.

---

### Note

You must specify the *RepType*. See:

- `SYMCLI_REPLICATE_HOP_TYPE=<RepType>`
- 

Set a nonzero value for either a *CycleTime* or a *Delay*time, (even though their default values are zero). See:

- `SYMCLI_REPLICATE_CYCLE=CycleTime`
- `SYMCLI_REPLICATE_CYCLE_DELAY=Delay`

## Format of the symreplicate options file

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>

```

## Set 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:

### `symreplicate` options file parameters

`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.

The default for *NumCycles* is 0, the *symreplicate* session cycles forever.

**SYMCLI\_REPLICATE\_CYCLE\_OVERFLOW**=*<OvfMethod>*

Describes what to do if the cycle overruns the specified *CycleTime*.

Valid values for *OvfMethod* are:

**IMMEDIATE**

(default) Begins next cycle immediately.

**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.

*TimeLimit* value controls how long *symreplicate* retries certain types of operations.

The default general *TimeLimit* is 00:30 if not specified.

A *TimeLimit* value of zero (0) indicates that no time limit applies, and the operation to be retries indefinitely.

*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

**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. *SleepTime* must be greater than zero (0).

The default value for *SleepTime* is 10 seconds.

*SleepTime* must be specified using one of the following formats:

**hh:mm**

Specifies *SleepTime* in number of hours and minutes.

**sss**

Specifies *SleepTime* in seconds.

**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.

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.

If not specified, the default factor is 4.

**SYMCLI\_REPLICATE\_TF\_CLONE\_EMULATION=***<TRUE|FALSE>*



---

**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.

---

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.

Indicates that TF/Clone emulation is enabled/disabled.

**FALSE**

(default) The TF/Clone emulation default is disabled.

**TRUE**

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.

**3 (default)**

Used if the `SYMCLI_REPLICATE_CONS_SPLIT_RETRY` option parameter is not specified when a consistent split (-consistent) is requested.

**0**

No retry attempts are 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. Valid values are:

**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>`

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. Values for *EstablishDelay*:

Range: Delay of 0 to 30 seconds

Default: 0

`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.

## Manage 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.

### Recover 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.

---

### Release 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.

#### Syntax

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.

### Restrictions

- 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 holder 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.

### Example

To release devices locks on a terminated session for device group prod on array 35002:

```
symreplicate -g prod release -sid 35002
```

## Acquire 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>.



# CHAPTER 12

## TimeFinder and SRDF operations

This chapter describes the following topics:

- [TimeFinder consistent splits in SRDF configurations](#)..... 398
- [Multi-hop operations](#) ..... 399
- [TimeFinder SnapVX and SRDF](#).....403

## 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 Engenuity 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.

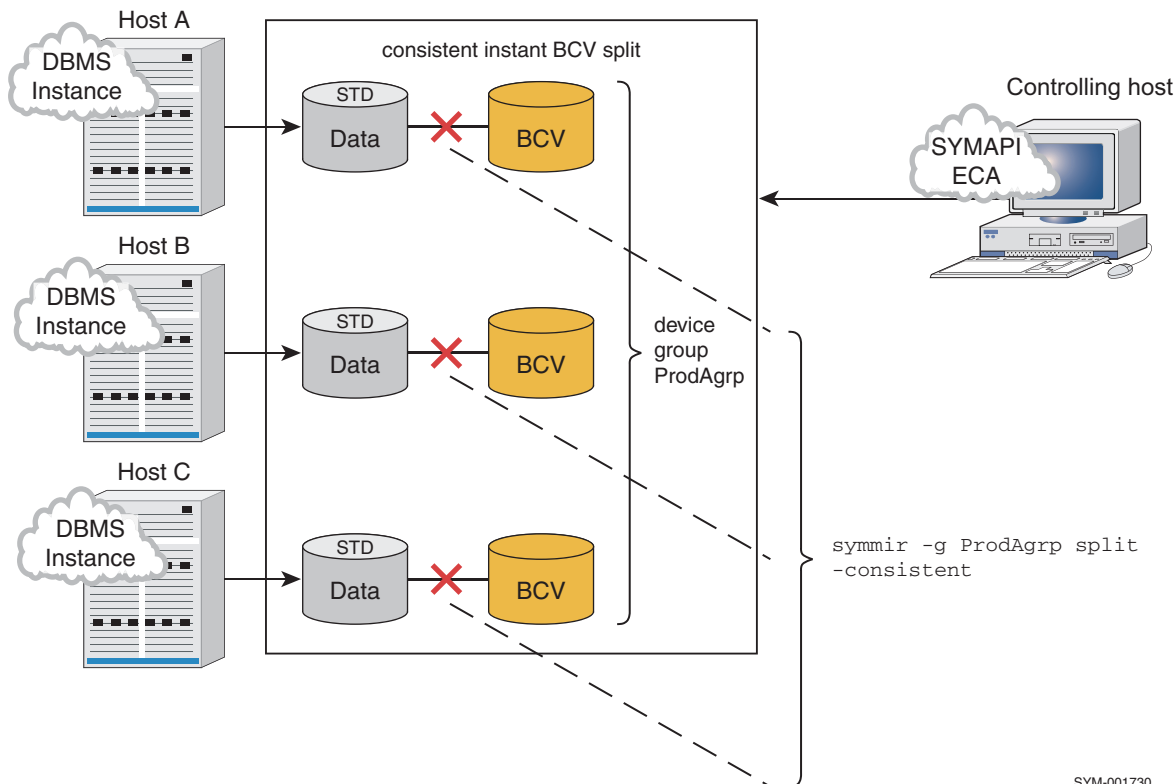
### Engenuity Consistency Assist

Use the `symir` command with `-consistent` option to perform TimeFinder consistent split operations.

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.

The following image shows how a control host can perform ECA consistent splits for three database hosts that access devices on a array.

**Figure 126** ECA consistent split



To consistently split BCV pairs using ECA:

SYM-001730

- 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 128 on page 402](#), 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 128 on page 402](#)), 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 128 on page 402](#):

- 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: preparing for multi-hop operations

`symmir` operations require an existing group of SRDF devices.

To create a device group containing STD and BCV RDF1 devices:

### Procedure

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:

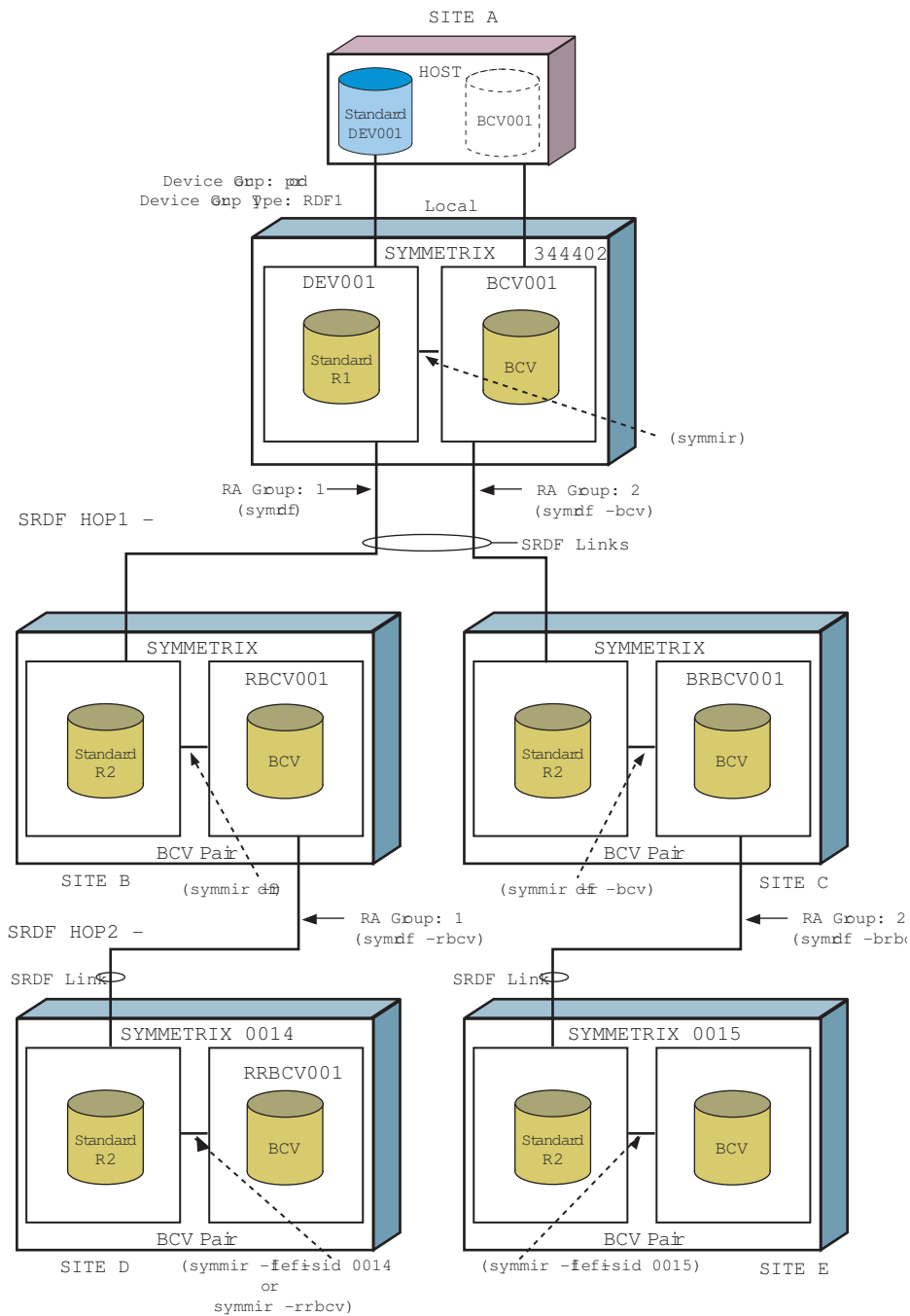
```
symdg -g prod add dev 0001 -sid 344402 DEV001
```

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 127** Commands used to perform splits in a complex configuration



## Control basic operations in a multi-hop configuration

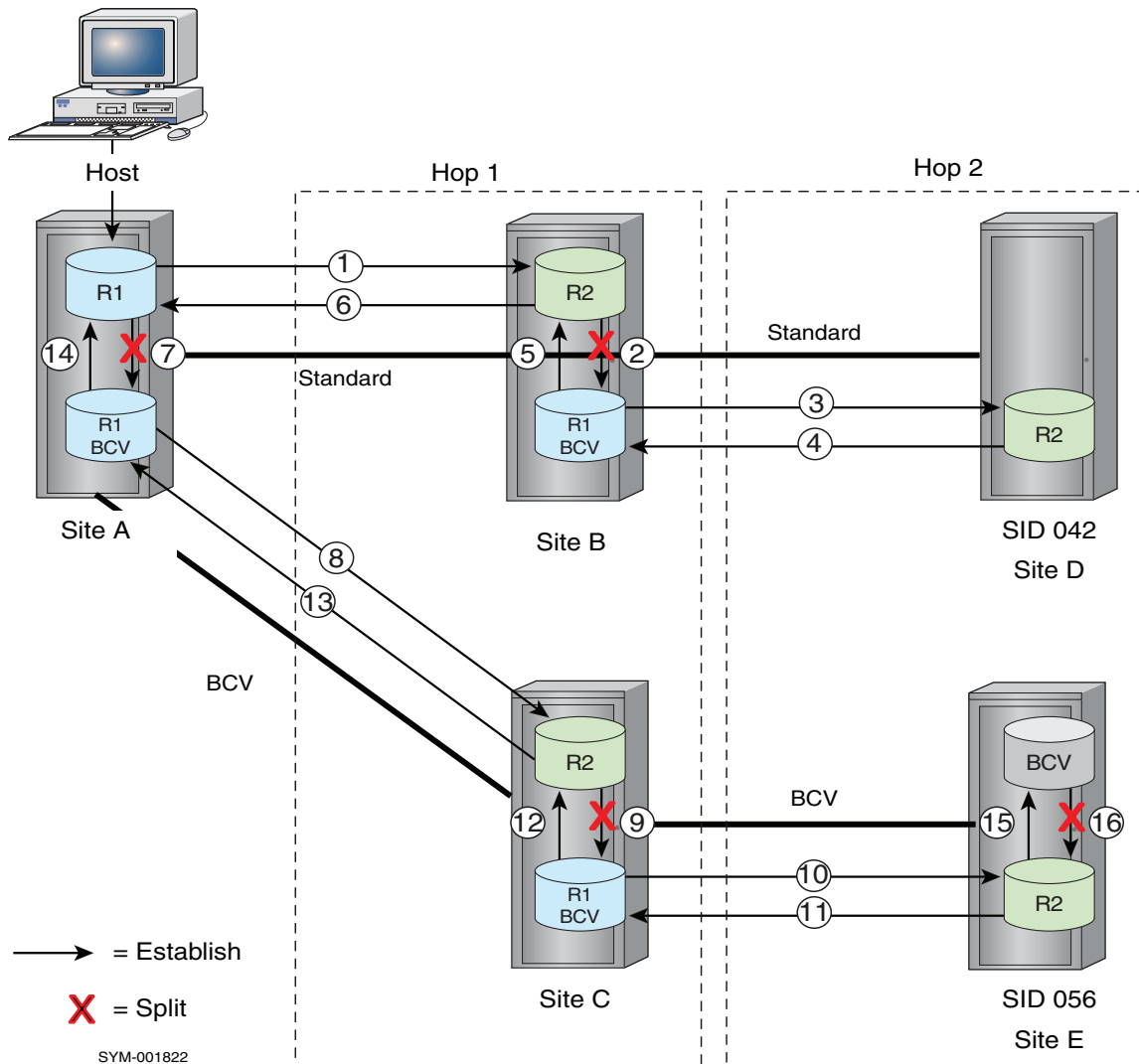
The following table lists the sequence of commands to perform basic control operations in a multi-hop configuration.

Each step number correlates to a bubble number in [Figure 128 on page 402](#).



**Table 43** Basic operations in a multi-hop configuration

| Step | CLI control operation                                                                                        | Description                                                            |
|------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| 1    | <code>symrdf -g &lt;&gt; establish</code>                                                                    | Creates the standard associated hop 1 copy.                            |
| 2    | <code>symmir -g &lt;&gt; split -rdf</code>                                                                   | Splits the standard associated hop 1 BCV device pair.                  |
| 3    | <code>symrdf -g &lt;&gt; establish -rbcv</code>                                                              | Creates the standard associated hop 2 copy.                            |
| 4    | <code>symrdf -g &lt;&gt; restore -rbcv</code>                                                                | Restores the standard associated hop 1 BCV with the hop 2 copy.        |
| 5    | <code>symmir -g &lt;&gt; restore -rdf</code>                                                                 | Restores the standard associated hop 1 copy with the hop 1 BCV.        |
| 6    | <code>symrdf -g &lt;&gt; restore</code>                                                                      | Restores the standard device with the hop 1 copy.                      |
| 7    | <code>symmir -g &lt;&gt; split</code>                                                                        | Splits the standard/BCV pair.                                          |
| 8    | <code>symrdf -g &lt;&gt; establish -bcv</code>                                                               | Creates the BCV associated hop 1 remote copy.                          |
| 9    | <code>symmir -g &lt;&gt; split -rdf -bcv</code>                                                              | Splits the BCV associated hop 1 device pair.                           |
| 10   | <code>symrdf -g &lt;&gt; establish -brbcv</code>                                                             | Creates the BCV associated hop 2 copy.                                 |
| 11   | <code>symrdf -g &lt;&gt; restore -brbcv</code>                                                               | Restores the BCV associated hop 1 BCV with the hop 2 copy.             |
| 12   | <code>symmir -g &lt;&gt; restore -rdf -bcv</code>                                                            | Restores the standard device associated hop 1 copy with the hop 1 BCV. |
| 13   | <code>symrdf -g &lt;&gt; restore -bcv</code>                                                                 | Restores the BCV device with the hop 1 copy.                           |
| 14   | <code>symmir -g &lt;&gt; restore</code>                                                                      | Restores the standard device with the BCV copy.                        |
| 15   | <code>symmir -f &lt;&gt; -sid 056 establish</code><br>or<br><code>symmir -g &lt;&gt; -rrbcv establish</code> | Creates the BCV associated hop 2 BCV copy.                             |
| 16   | <code>symmir -f &lt;&gt; -sid 056 split</code><br>or<br><code>symmir -g &lt;&gt; -rrbcv</code>               | Splits the BCV-associated hop 2 device pair.                           |

**Figure 128** Basic operations in multi-hop SRDF configurations

## System-wide split commands

[Figure 128 on page 402](#) 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.

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.

**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
```

## 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.

#### NOTICE

Starting in Solutions Enabler 8.0.2/HYPERMAX OS Q1 2015 SR you can manage SRDF operations using storage groups.

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 steps create 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:

```
symmdg create DeptAB -type ANY
symmdg -g DeptAB -sid 000197300076 addall dev -devs 00019:0001A
symmdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt
```

2. Add devices 0001D:0001E on remote array (R21, 1st hop) as TGT devices.

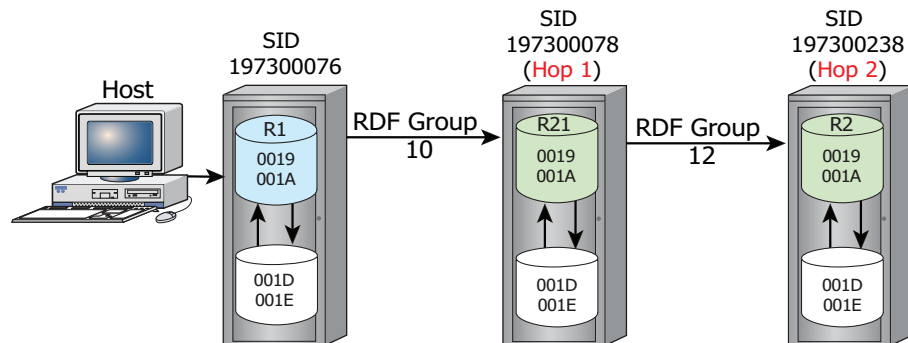
```
symmdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt -rdf
```

3. Add devices 0001D:0001E in RDF group 12 on remote array (R2, 2nd hop) as TGT devices:

```
symmdg -g DeptAB -sid 000197300076 addall dev -devs 0001D:0001E -tgt -rdf
```

The following image shows the resulting configuration:

**Figure 129** SnapVX and Cascaded SRDF



### Examples

The following examples use the configuration shown in the preceding image:

- 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 steps create composite groups and add devices 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
```

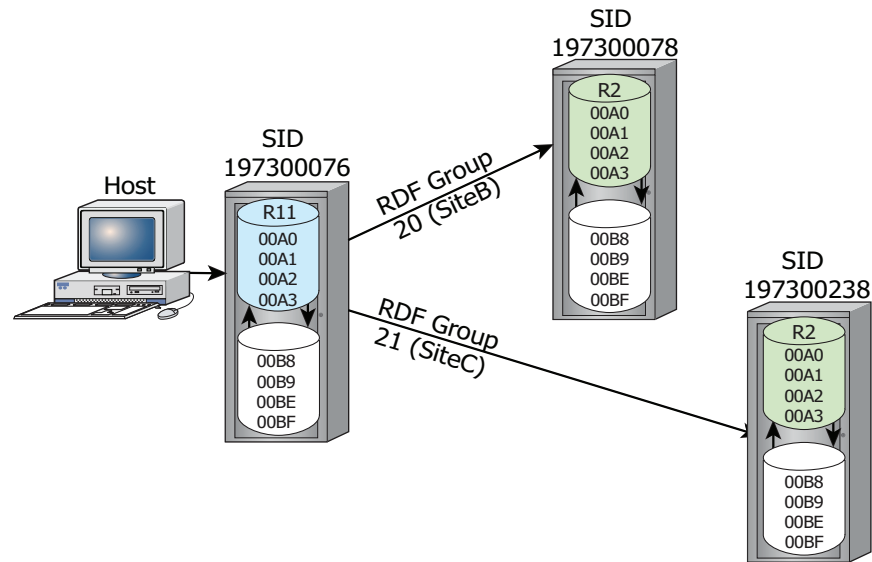
4. Add devices B8:BF in RDF group 20 as RTGTs:

```
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
```

The following image shows the resulting configuration:

**Figure 130** SnapVX and Concurrent SRDF**Examples**

The following examples use the configuration shown in the preceding image.

- 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 (named SiteCSnap) 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 13

## SRDF Automated Recovery Operations

This chapter describes the following topics:

- [Automated Recovery overview](#) ..... 408
- [Launch SRDF Automated Recovery](#) ..... 410
- [Stop SRDF Automated Recovery](#) ..... 413
- [symrecover options file parameters](#) ..... 413

## Automated Recovery 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.

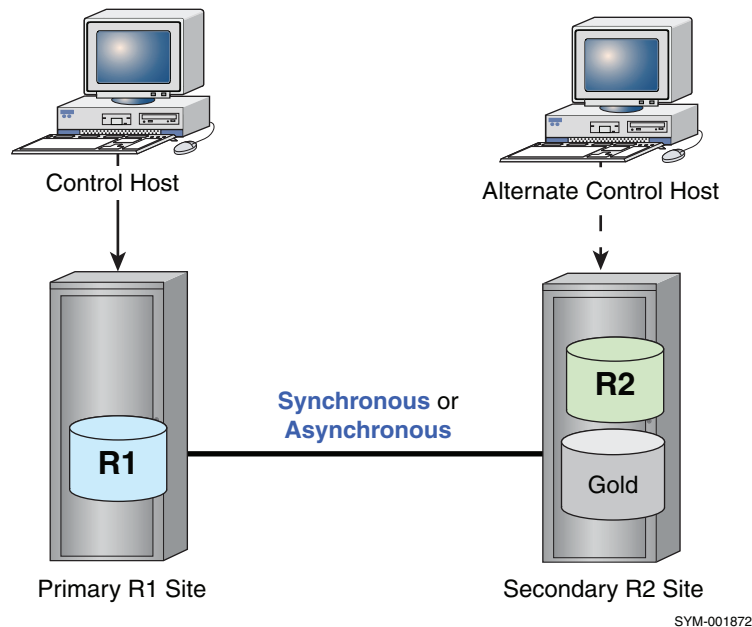
If SRDF Automated Recovery detects a session failure, it attempts an automatic recovery and restart of the session using the `symrecover` command. The restart uses the pre-configured settings specified in the `symrecover` options file.

This options file provides parameters for:

- Email notification for error logging and events
- Actions for monitoring, recovery, and restart

The following image shows a basic SRDF recovery environment:

**Figure 131** 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.



---

**Note**

SRDF Automated Recovery is not supported in SRDF/Star environments.

---

## SRDF Automated Recovery restrictions

- The `symrecover` session must be started either at the primary R1 site or the remote R2 site.
- If the group is concurrent, then `symrecover` must be run from the R1 workload site.
- SRDF/Metro configurations cannot be monitored by SRDF Automated Recovery.
- 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.
- 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 using device groups, `symrecover` can be started at other sites.

---

### Gold copy restrictions

- R2 gold copying can be performed with either native clones or Business Continuity Volumes (BCVs).
- BCV-to-STD association for the R2 gold copy is dynamic using the `symmir` defaults.

### Restart restrictions

---

**Note**

[symrecover options file parameters on page 413](#) provides 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_startup` parameters are specified, and an R22 device has another SRDF mirror that is already RW on the link.

## 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 44 on page 413](#) 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_startup` parameters are specified, and an R22 device has another SRDF mirror that is already RW on the link.

# Launch SRDF Automated Recovery

Use the `symrecover` command to launch and optimize SRDF Automated Recovery. The `symrecover` command can be run 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 to 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.

---

**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 an array running Enginuity 5875 or 5876 or HYPERMAX OS 5977. The standard device must be on an array running Enginuity 5671, 5773.50154, 5875, or 5876.

---

The `symrecover` command can be invoked manually from the command line, is typically configured to run continuously in the background using one of the following:

- Windows Scheduled Tasks
- UNIX CRON/scheduled task
- UNIX (RC.2) file

**Syntax**

Use the following syntax to launch SRDF Automated Recovery operations:

```
symrecover [-h]
```

```
symrecover [-env | -version]

symrecover start {-g DgName | -cg CgName}
[-mode {SYNC | ASYNC}] [-out LogPath]
[-options OptnFile]
```

## Options

### Note

Either a device group (`-g DgName`) or composite group (`-cg CgName`) must be specified.

- `-g DgName`  
Specifies a device group.
- `-cg CgName`  
Specifies a composite group.
- `-mode {SYNC | ASYNC}`  
Specifies the SRDF session type, either synchronous or asynchronous. There is no default; this option must be specified.
- `-out LogPath`  
Specifies an alternate fully-qualified directory location for the log file.
- `-options OptnFile`  
Specifies the fully-qualified file containing program options. See [symrecover options file parameters on page 413](#) for a list of possible settings.

## Restrictions

- You can define devices in groups on the R2 side with a corresponding partner but `symrecover` cannot start in this environment. You cannot monitor groups on the R2 side when the remote partner is concurrent. You must monitor these groups from the host.
- 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.

## 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
```

To manually start recovery for an SRDF/A composite group named `RDFAmn`, 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
```

## Recover cascaded SRDF

### Syntax

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
```

### Options

**cascaded\_monitor\_both\_hops = 1**

Allows recovery on both hops.

**goldcopy\_location = All**

Builds 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.

## Stop 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

The following table describes the valid settings in the `symrecover` options file.

**Table 44** symrecover options file parameters

| Setting                                                            | Description                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cascaded_monitor_both_hops= [0 1]</code>                     | 0 - (default) Monitors/recovers a single hop only invoked from any site.<br>1 - For cascaded SRDF environments, the <code>symrecover</code> session monitors both hops linking the cascaded sites.<br><br>The <code>symrecover</code> session ignores the <code>-mode</code> option and can be invoked at either the R1 primary or the remote R2 tertiary site only (not at R21 site). |
| <code>email_addr_target= &lt;e_addr1, e_addr2, ..., ...&gt;</code> | Email notification address on errors. If any of the <code>email_*</code> options are specified, then this option must also be specified to activate email alerts.<br>Multiple comma delimited addresses may be specified. There is no default value.                                                                                                                                   |
| <code>email_addr_source= e_addr1</code>                            | Specifies an address that will be used as the 'from' field for any e-mails that <code>symrecover</code> sends.<br>No validity checks are done for the e-mail address. If this setting is not specified, then a default value is generated based on the array's hostname and current user account.                                                                                      |
| <code>email_server= e_srvr_addr</code>                             | Specifies the host target email server.<br>If any of the <code>email_*</code> options are specified then this option must also be specified to activate email alerts. There is no default value.                                                                                                                                                                                       |
| <code>email_subject= err_subject_string</code>                     | Specifies the email notification subject on errors. The default value is: SymRecover Alert: Host [HostName] Group [GrpName]                                                                                                                                                                                                                                                            |
| <code>email_log_level= SeverityLevel</code>                        | The severity level desired for the email alert triggering message. Valid values are:<br>0 = Off<br>1 = Only errors are reported<br>2 = Errors and warnings are reported<br>3 = Errors, warnings, and informational messages are reported<br>4 = All messages are reported, including all SYMCLI commands and responses                                                                 |

**Table 44** symrecover options file parameters (continued)

| Setting                                                                                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                     | <p><b>Note</b></p> <p>For each message that meets the particular logging level requirement, an email is sent with that message. It is highly recommended to set the severity level to either a 1 or a 2.</p> <p>If the required email options (<code>email_server</code> and <code>email_addr_target</code>) are not specified, the default value is 0. If they are specified, the default value is 1.</p>                                                                                                                                                                                                                                               |
| <code>goldcopy_location= LocationValue</code>                                                                                                       | <p>Specifies the location of the backup gold copy. Valid (case-insensitive) values are:</p> <p>NONE = No gold copy is desired. All other gold copy optional parameters in this list are ignored.</p> <p>R2 = A gold copy on the R2 site is desired. This is the default setting. Any R2 BCV pairs must already defined before calling <code>symrecover</code>.</p>                                                                                                                                                                                                                                                                                       |
| <code>goldcopy_type=CopyType</code><br>Old alternate, if still necessary:<br><code>goldcopy_type_r2= CopyType</code>                                | <p>Specifies the type of gold copy to create on the R2 side. Valid (case-insensitive) values are:</p> <p>NONE = No gold copy is desired. All other <code>goldcopy_*</code> options are ignored.</p> <p>BCV = BCV gold copy on the R2 side is created. This is the default.</p> <p>CLONE = Clone gold copy on the R2 is created.</p> <p><b>Note</b></p> <p>For the BCV gold copy, the R2 BCVs must be paired with the R2 devices before starting <code>symrecover</code>. For the clone gold copy, the target devices must have a clone session with the R2 devices before starting <code>symrecover</code>.</p>                                          |
| <code>goldcopy_state_startup= CopyType</code><br>Old alternate, if still necessary:<br><code>goldcopy_bcv_r2_mirror_state_startup= CopyState</code> | <p>Specifies the desired state of the R2 gold copy upon routine startup. Valid (case-insensitive) values are:</p> <p>ESTABLISH = The devices must be established (BCV gold copy only).</p> <p>SPLIT = The devices must be split (BCV gold copy only).</p> <p>ACTIVATED = The devices must be in the copied state (clone gold copy only).</p> <p>CREATED = The devices must be in the precopy state (clone gold copy only).</p> <p>NONE = The devices must be unchanged. This is the default.</p> <p><b>Note</b></p> <p>If the gold copy type is BCV and the default state of the BCVs is ESTABLISH, this is likely to increase SRDF/A session drops.</p> |
| <code>goldcopy_state_post_restart= CopyState</code><br>Old alternate, if still necessary:                                                           | <p>Following a successful SRDF/A session restart or BCV resync, specifies which state the R2 gold copy should be. Valid (case-insensitive) values are:</p> <p>ESTABLISH = The devices must be left established (BCV gold copy only).</p>                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Table 44** symrecover options file parameters (continued)

| Setting                                                                                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| goldcopy_bcv_r2_mir_state_post_restart= CopyState                                                                                | <p>SPLIT = The devices must be split, which is the default (BCV gold copy only).</p> <p>ACTIVATED = The devices must be in the copied state (clone only).</p> <p>CREATED = The devices must be in the precopy state (clone only).</p> <hr/> <p><b>Note</b></p> <p>If the gold copy type is BCV and the default state of the BCVs is ESTABLISH, this is likely to increase SRDF/A session drops.</p>                                                                                                                                                                                                         |
| goldcopy_max_wait= MaxWaitTime<br>Old alternate, if still necessary:<br>goldcopy_max_wait_bcv= MaxWaitTime                       | <p>Specifies the length of time, in seconds, for <code>symrecover</code> to wait for synchronization. Valid values are 0 to <b>maxint</b> (2147483647). The default is 0, which indicates for <code>symrecover</code> to wait forever.</p> <p>For clone gold copies, if the <code>goldcopy_state_post_restart</code> option is set to <b>activated</b>, it waits for the clone copied state to be reached before performing synchronization. If this option is set to <b>created</b>, it waits for the clone precopied state to be reached.</p>                                                             |
| goldcopy_resync_interval= resynctime<br>Old alternate, if still necessary:<br>goldcopy_bcv_r2_mirror_resync_interval= resynctime | <p>Defines the resync interval, in minutes, for <code>symrecover</code> 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.</p> <p>Valid values are 0, and 15 to <b>maxint</b>. Zero (0) indicates that the mirrors are never to be automatically synchronized outside of error-producing events. The default setting is 15.</p> <hr/> <p><b>Note</b></p> <p>If the gold copy type is BCV, the act of frequently synchronizing the R2 BCVs is likely to increase SRDF/A session drops.</p> |
| goldcopy_clone_list= List                                                                                                        | <p>For a clone gold copy, this option tells <code>symrecover</code> which list within the device group or the composite group to search for clone devices.</p> <p>Valid (case-insensitive) values are:</p> <p>TGT = Uses the TGT list.</p> <p>BCV = Uses the BCV list.</p>                                                                                                                                                                                                                                                                                                                                  |
| monitor_cycle_time= cycletime                                                                                                    | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| monitor_only= [0 1]                                                                                                              | <p>Specifies to only monitor the state of specified group. No recovery actions will take place. Valid values are:</p> <p>0 = Disable the option. This is the default.</p> <p>1 = Enable monitoring.</p> <hr/> <p><b>Note</b></p> <p><code>monitor_only</code>, <code>run_once</code>, and <code>run_until_first_failure</code> are mutually exclusive options.</p>                                                                                                                                                                                                                                          |

**Table 44** symrecover options file parameters (continued)

| Setting                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>run_once= [0 1]</code>                          | <p>Specifies to check the status of the group once. If the group needs recovery actions perform them. Exit after one check. This option ignores the setting of <code>restart_max_attempts</code>. Valid values are:</p> <p>0 = Disable the option. This is the default.</p> <p>1 = Enable status check.</p> <hr/> <p><b>Note</b></p> <p><code>monitor_only</code>, <code>run_once</code>, and <code>run_until_first_failure</code> are mutually exclusive options.</p>                                                                                                                                                                                                                                                                                                                                 |
| <code>run_until_first_failure= [0 1]</code>           | <p>Specifies to monitor the group until the first failure occurs and then exit without performing any recovery action. This option ignores the setting of <code>restart_max_attempts</code>. Valid values are:</p> <p>0 = Disable the option. This is the default.</p> <p>1 = Enable monitoring.</p> <hr/> <p><b>Note</b></p> <p><code>monitor_only</code>, <code>run_once</code>, and <code>run_until_first_failure</code> are mutually exclusive options.</p>                                                                                                                                                                                                                                                                                                                                        |
| <code>rdfg= rdfgvalue</code>                          | <p>When working with device groups or composite groups that contain concurrent devices, <code>symrecover</code> supports monitoring only one of the SRDF groups that contain mirrors of the concurrent devices. Use the <code>rdfg</code> option to indicate the SRDF group that <code>symrecover</code> should monitor. Note that monitoring of concurrent SRDF defined groups is only supported when <code>symrecover</code> 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.</p> <hr/> <p><b>Note</b></p> <p>If the group is a composite group, and consistency is enabled, this must be of the "name:" format and this value is case sensitive.</p> |
| <code>restart_adcopy_resynch_threshold= tracks</code> | 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>restart_attempt_pause= time</code>              | <p>Inserts a specified wait time before an attempt is made to restart a failed session to allow for things to settle down. After the <code>restart_attempt_pause</code> is complete, <code>symrecover</code> redrives the overall monitor loop. If there is still a problem, the restart failure count is incremented and a restart is attempted.</p> <p>Valid values are 30 to 3600 seconds. The default is 60 seconds.</p>                                                                                                                                                                                                                                                                                                                                                                           |
| <code>restart_delay= time</code>                      | <p>Inserts a specified wait time after an attempt is made to restart a failed session and the attempt itself fails.</p> <p>Valid values are 0 (no delay, immediately restart) to maxint. The default is 30 seconds.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



**Table 44** symrecover options file parameters (continued)

| Setting                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>restart_group_on_startup= [0 1]</code>                     | On <code>symrecover</code> startup, if the group being monitored is not initially in a Consistent state (for SRDF/A) or a Synchronized state (for SRDF/S), <code>symrecover</code> considers this an error condition and exits. If this option is specified, <code>symrecover</code> will attempt to recover the group on startup. Valid values are:<br>0 = Disable the option. This is the default.<br>1 = Attempt recovery on startup. |
| <code>restart_max_attempts= attempts</code>                      | Specifies the maximum number of restart attempts that are performed within the <code>restart_window</code> interval. After this limit is reached the program terminates.<br>The range is from 0 to <code>maxint</code> . The value of 0 specifies to attempt indefinitely. The default value is 5 attempts.                                                                                                                              |
| <code>restart_max_wait_adcopy_sync= time</code>                  | Specifies the length of time (in seconds) during a restart for a program to wait for a group to achieve the <code>restart_adcopy_resync_threshold</code> number of tracks pending. Valid values are 0 to <code>maxint</code> . The value of 0 specifies to wait forever. The default is 0.                                                                                                                                               |
| <code>restart_max_wait_state_change= statetime</code>            | 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 <code>maxint</code> . The value of 0 specifies to wait forever. The default is 0.                                                                                                                                                                                      |
| <code>restart_max_wait_warn_interval= warntime</code>            | Specifies the length of time (in seconds) to display a progress warning message while waiting for a state change to occur during a restart. Valid values are 0 and 30 to <code>maxint</code> . The value of 0 specifies to wait forever. The default is 600 seconds.                                                                                                                                                                     |
| <code>restart_rdfa_min_cycle_warn_interval= cyclewarntime</code> | Specifies the length of time (in seconds) before repetitively displaying a warning when the RDFA minimum cycle time exceeds the <code>restart_rdfa_min_cycle_warn_value</code> parameter. Valid values are 30 to <code>maxint</code> . The default is 600.                                                                                                                                                                               |
| <code>restart_rdfa_min_cycle_warn_value= warntime</code>         | 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.<br>Valid values are 0 and 30 to <code>maxint</code> . The value of 0 means this feature is turned off, which is the default.                                                                                                                                               |
| <code>restart_state_syncinprog_wait_time time</code>             | The maximum length of time (in seconds) during a group <code>syncinprog</code> state that sleep is done before rechecking the group status. Valid values are [30] to [maxint]. The default is [120] seconds.                                                                                                                                                                                                                             |
| <code>restart_state_transmit_warn_interval= time</code>          | Specifies the interval of time (in seconds) that while a group remains in a transmit idle state, to generate a warning message.<br>Valid values are 0 to <code>maxint</code> . The default is 300 seconds.                                                                                                                                                                                                                               |
| <code>restart_state_transmit_wait_time= transwaittime</code>     | 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 <code>maxint</code> . The default is 120 seconds.                                                                                                                                                                                                                  |

**Table 44** symrecover options file parameters (continued)

| Setting                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>restart_sync_type= synctype</code> | <p>Specifies the type of synchronization to be used following the detection of a failed SRDF/A session. Valid values are:</p> <p>ADCOPY = adaptive copy disk (default).</p> <p>SYNC = synchronous mode.</p> <p>NONE = No intermediate track resynch stage will be attempted. A direct re-establish using the existing SRDF session mode will be attempted.</p> <p>Note that if <code>cascaded_monitor_both_hops</code> is set, <code>restart_sync_type</code> is ignored as ADCOPY is used in the R21-&gt;R2 link at restart.</p> |
| <code>restart_window= time</code>        | <p>Specifies a time window (in seconds) during which no more than <code>restart_max_attempts</code> failures and accompanying restart attempts will be tolerated before monitoring is terminated. The window begins at the time of the first failure and ends <code>restart_window</code> seconds later. A new window begins with a failure after expiration of the previous window.</p>                                                                                                                                          |
| <code>log_level= level</code>            | <p>The desired logging level. Valid values are:</p> <p>0 = Off</p> <p>1 = Only errors are reported</p> <p>2 = Errors and warnings are reported</p> <p>3 = Errors, warnings, and informational messages are reported (default)</p> <p>4 = All messages are reported</p>                                                                                                                                                                                                                                                            |

# APPENDIX A

## SRDF operations and pair states

This appendix describes the following topics:

- [SRDF operations and applicable pair states](#) .....420
- [Cascaded SRDF control operations and applicable pair states](#)..... 424
- [Cascaded SRDF set operations and applicable pair states](#)..... 430
- [Concurrent SRDF operations and applicable pair states](#).....432
- [Consistency group operations and applicable pair states](#) ..... 438

## SRDF operations and applicable pair states

When a command to perform an SRDF control operation is issued, 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.

The `-force` option is required for devices that are part of an SRDF/Metro configuration.

The `-force` option is required for restore, update R1, and failback operations for devices that are running in SRDF/A mode.

### 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 in the SyncInProg state:

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

### NOTICE

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

In the following table, the first column lists the control operations that can be invoked for the listed pair states.

Allowed actions are noted by Ys.

The **Partitioned1** 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** SRDF control operations and applicable pair states

| Control operation | R1 -> R2 pair state |              |         |           |             |              |              |            |              |         |            |              |              |            |
|-------------------|---------------------|--------------|---------|-----------|-------------|--------------|--------------|------------|--------------|---------|------------|--------------|--------------|------------|
|                   | Sync in prog        | Synchronized | Split   | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid | Consistent | TransmitIdle | ActiveActive | ActiveBias |
| deletepair        |                     |              | Y,a,b,c | Y,a,b,c   | Y,a,b,c     |              |              |            |              |         |            |              |              |            |
| half_deletepair   |                     |              | Y,a,b,c | Y,a,b,c   | Y,a,b       | Y,a,b,c      | Y,a,b,c      |            |              |         |            |              |              |            |
| movepair          |                     |              | Y       | Y         | Y           |              |              |            |              |         |            |              |              |            |

Table 45 SRDF control operations and applicable pair states (continued)

| Control operation   | R1 -> R2 pair state    |                      |                        |                      |                      |                      |                      |                      |                        |                        |                      |                      |              |            |
|---------------------|------------------------|----------------------|------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|------------------------|------------------------|----------------------|----------------------|--------------|------------|
|                     | Sync in prog           | Synchronized         | Split                  | Suspended            | Failed over          | Partitioned1         | Partitioned2         | R1 updated           | R1 upinprog            | Invalid                | Consistent           | TransmitIdle         | ActiveActive | ActiveBias |
| half_movepair       |                        |                      | Y                      | Y                    | Y                    | Y                    | Y                    |                      |                        |                        |                      |                      |              |            |
| swap                |                        |                      | γ <sub>d</sub>         | γ <sub>d,e</sub>     | γ <sub>d</sub>       |                      |                      | γ <sub>d</sub>       |                        |                        |                      |                      |              |            |
| swap -refresh R1    |                        |                      | γ <sub>d,f</sub>       | γ <sub>d,e,f</sub>   | γ <sub>d,f</sub>     |                      |                      | γ <sub>d,f</sub>     |                        |                        |                      |                      |              |            |
| swap -refresh R2    |                        |                      | γ <sub>d,f</sub>       | γ <sub>d,e,f</sub>   | γ <sub>d,f</sub>     |                      |                      |                      |                        |                        |                      |                      |              |            |
| half_swap           |                        |                      | γ <sub>d</sub>         | γ <sub>d</sub>       | γ <sub>d</sub>       | γ <sub>d</sub>       | γ <sub>d</sub>       |                      |                        |                        |                      |                      |              |            |
| establish           |                        |                      | γ <sub>g</sub>         | γ <sub>g</sub>       | γ <sub>g,h</sub>     |                      |                      | γ <sub>g,h</sub>     |                        | γ <sub>i</sub>         |                      |                      |              |            |
| establish -full     |                        |                      | γ <sub>g</sub>         | γ <sub>g</sub>       | γ <sub>g,h</sub>     |                      |                      | γ <sub>g,h</sub>     |                        | γ <sub>i,j</sub>       |                      |                      |              |            |
| split               | γ <sub>a,b,c,k,l</sub> | Y                    |                        | γ <sub>a,b,c,l</sub> |                      |                      |                      | γ <sub>a,b,c,l</sub> |                        |                        | γ <sub>a,b,c,l</sub> |                      |              |            |
| restore             |                        |                      | γ <sub>f</sub>         | γ <sub>f,n</sub>     | γ <sub>f,h</sub>     |                      |                      | γ <sub>f,h</sub>     |                        | γ <sub>m</sub>         |                      |                      |              |            |
| restore -full       |                        |                      | γ <sub>f</sub>         | γ <sub>f,n</sub>     | γ <sub>f,h</sub>     |                      |                      | γ <sub>f,h</sub>     |                        | γ <sub>m,o</sub>       |                      |                      |              |            |
| update              |                        |                      |                        | γ <sub>a,f,p,q</sub> | γ <sub>a,f,q</sub>   |                      |                      | γ <sub>a,f,q</sub>   |                        |                        |                      |                      |              |            |
| failback            |                        |                      | γ <sub>a,f,g</sub>     | γ <sub>a,f,g,p</sub> | γ <sub>a,f,g</sub>   | γ <sub>g,r,s</sub>   | γ <sub>g,r,s</sub>   | γ <sub>a,f,g</sub>   | γ <sub>a,f,g</sub>     |                        |                      |                      |              |            |
| failover            | γ <sub>k,l,t,u,v</sub> | γ <sub>l,t,u,v</sub> | γ <sub>l,s,t,u,v</sub> | γ <sub>l,t,u,v</sub> |                      | γ <sub>l,u,v,w</sub> | γ <sub>l,u,v,w</sub> | γ <sub>l,t,u,v</sub> | γ <sub>k,l,t,u,v</sub> | γ <sub>l,s,t,u,v</sub> | γ <sub>l,t,u,v</sub> | γ <sub>l,t,u,v</sub> |              |            |
| failover -establish | γ <sub>d,f,j,k,t</sub> | γ <sub>d,f,t</sub>   | γ <sub>d,l,s,t</sub>   | γ <sub>d,f,t</sub>   | γ <sub>d,f,t</sub>   |                      |                      | γ <sub>d,f,t</sub>   | γ <sub>d,f,k,t</sub>   | γ <sub>d,f,s,t</sub>   |                      |                      |              |            |
| failover -restore   | γ <sub>d,g,k,x</sub>   | γ <sub>d,g,x</sub>   | γ <sub>d,g,s,x</sub>   | γ <sub>d,g,x</sub>   | γ <sub>d,g,s,x</sub> |                      |                      |                      |                        | γ <sub>d,g,s,x</sub>   | γ <sub>d,g,x</sub>   |                      |              |            |
| invalidate R1       |                        |                      |                        | γ <sub>f,y</sub>     |                      |                      |                      |                      |                        |                        |                      |                      |              |            |
| invalidate R2       |                        |                      |                        | γ <sub>g</sub>       |                      |                      |                      |                      |                        |                        |                      |                      |              |            |
| merge               |                        |                      |                        | γ <sub>n,z,aa</sub>  | γ <sub>s,z,aa</sub>  |                      |                      |                      |                        |                        |                      |                      |              |            |
| msc_cleanup         |                        |                      | Y                      | Y                    | Y                    | γ <sub>w</sub>       | γ <sub>w</sub>       |                      |                        |                        |                      |                      |              |            |
| not_ready R1        | γ <sub>e,ab</sub>      | γ <sub>e,ab</sub>    | γ <sub>e,ab</sub>      | γ <sub>e,ab</sub>    | γ <sub>ab</sub>      | γ <sub>e,r,a,b</sub> | γ <sub>e,r,a,b</sub> | γ <sub>ab</sub>      | γ <sub>ab</sub>        | γ <sub>e,ab</sub>      | γ <sub>e,ab</sub>    | γ <sub>e,r,ab</sub>  |              |            |
| not_ready R2        | γ <sub>j,ab,ac</sub>   | γ <sub>ab,ac</sub>   | γ <sub>ab,a,c</sub>    | γ <sub>ab</sub>      | γ <sub>ab,ac</sub>   | γ <sub>ab,w,ad</sub> | γ <sub>ab,w,ad</sub> | γ <sub>ab,ac</sub>   | γ <sub>ab,ac</sub>     | γ <sub>j,ab,a,c</sub>  |                      |                      |              |            |
| ready R1            | γ <sub>ab</sub>        | γ <sub>ab</sub>      | γ <sub>ab</sub>        | γ <sub>ab</sub>      | γ <sub>ab</sub>      | γ <sub>r,ab</sub>    | γ <sub>r,ab</sub>    | γ <sub>ab</sub>      | γ <sub>ab</sub>        | γ <sub>ab</sub>        | γ <sub>ab</sub>      | γ <sub>ab</sub>      |              |            |
| ready R2            | γ <sub>j,ab</sub>      | γ <sub>ab</sub>      | γ <sub>ab</sub>        | γ <sub>ab</sub>      | γ <sub>ab</sub>      | γ <sub>ab,w</sub>    | γ <sub>ab,w</sub>    | γ <sub>ab</sub>      | γ <sub>ab</sub>        | γ <sub>j,ab</sub>      |                      |                      |              |            |
| refresh R1          |                        |                      |                        | γ <sub>p,ae,af</sub> | γ <sub>s,af</sub>    |                      |                      |                      |                        |                        |                      |                      |              |            |

**Table 45** SRDF control operations and applicable pair states (continued)

| Control operation               | R1 -> R2 pair state |              |                  |                |               |              |              |            |              |             |             |                |              |            |
|---------------------------------|---------------------|--------------|------------------|----------------|---------------|--------------|--------------|------------|--------------|-------------|-------------|----------------|--------------|------------|
|                                 | Sync in prog        | Synchronized | Split            | Suspended      | Failed over   | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid     | Consistent  | TransmitIdle   | ActiveActive | ActiveBias |
| refresh R2                      |                     |              |                  | γg,p,s         |               |              |              |            |              |             |             |                |              |            |
| resume                          |                     |              |                  | Y              |               |              |              |            |              |             |             |                |              |            |
| suspend                         | γa,b,c,k,l,ag       | Y            | γa,b,c,l,s,ag,ah | γa,b,c,l,ad,ag | γa,b,c,h,l,ag | γa,b,c,l,ag  | γa,b,c,ag    | γa,b,c,ag  | γa,b,c,ag    | γa,b,c,k,ag | γa,b,c,l,ag | γa,b,c,l,ae,ag | γs           | γs         |
| disable                         | γu                  | γu           | γu               | γu             | γu            | γs,u         | γs,u         | γu         | γu           | γu          | γu          | γai            |              |            |
| enable                          | γaj                 |              | γaj              | γaj            | γaj           |              |              | γaj        | γaj          | γaj         | Y           | γai            |              |            |
| rw_disable R2                   | γj                  | Y            | γak              | Y              | γak           | γw,ad        | γw,ad        | γak        | γak          | γj          |             |                |              |            |
| rw_enable R1                    | γp                  | γp           |                  | γp             | γs            | γp,r         | γp,r         |            |              | γp          | γp          | γp,r           |              |            |
| rw_enable R2                    |                     |              |                  | γad            |               | γw,ad        | γw,ad        |            |              |             |             |                |              |            |
| write_disable R1                | γs                  | γs           | γ,s              | γs             |               | γr,al        | γr,al        |            |              | γs          | γs          | γr,al          |              |            |
| write_disable R2                |                     |              | γs               | γs             | γs            | γac,w        | γac,w        | γs         | γs           | γs          |             | γac,w          |              |            |
| activate - rdfa_dse             | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γap            |              |            |
| deactivate - rdfa_dse           | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γap            |              |            |
| activate - rdfa_devpace         | γam,an              |              |                  |                |               |              |              |            |              |             | γan         | γan,ap         |              |            |
| deactivate - rdfa_devpace       | γam,an              |              |                  |                |               |              |              |            |              |             | γan         | γan,ao,ap      |              |            |
| activate - rdfa_pace            | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γap            |              |            |
| deactivate - rdfa_pace          | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γao,ap         |              |            |
| activate - rdfa_wpace           | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γap            |              |            |
| deactivate - rdfa_wpace         | γam                 |              |                  |                |               |              |              |            |              |             | Y           | γap            |              |            |
| activate - rdfa_wpace_exe mpt   | Y                   | Y            | Y                | Y              | Y             |              |              | Y          | Y            |             | Y           |                |              |            |
| deactivate - rdfa_wpace_exe mpt | Y                   | Y            | Y                | Y              | Y             |              |              | Y          | Y            |             | Y           |                |              |            |

**Table 45** SRDF control operations and applicable pair states (continued)

- a. If remote invalid tracks are on the source side, must use -force.
- b. If there are local invalid tracks on the source side, must use -symforce if the source is not an R11 or R21.
- c. If there are local invalid tracks on the target side, must use -symforce if the target is not an R11 or R21.
- d. Not allowed if enabled for SRDF consistency protection.
- e. Write Disabled on the source.
- f. No local invalid tracks on the target side.
- g. No local invalid tracks on the source side.
- h. Source is not visible to any host.
- i. 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.
- j. Not allowed when SRDF/A is active.
- k. Can use -symforce.
- l. If enabled for SRDF consistency protection, must use -force.
- m. Source and target are Not Ready but the SRDF link is Ready and there are no remote invalid tracks on the source side.
- n. SA is Write Disabled, or is Not Ready on the source side, or must use -force.
- o. 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.
- p. SA is Write Disabled or is Not Ready on the source side.
- q. Not allowed if the R1 array is running Enginuity 5876 and the R2 array is running Enginuity 5977 or higher.
- r. Host application running while connected to the source.
- s. Must use -force.
- t. If remote invalid tracks are on the source side, must use -symforce.
- u. If enabled for CG SRDF consistency protection, must use -force.
- v. If local invalid tracks are on the target side, must use -symforce.
- w. Host application running while connected to the target.
- x. If remote invalid tracks are on the target side, must use -force.
- y. SA is Write Disabled, or is Not Ready on the source side, or must use -nowd.
- z. Source device is Read Write Enabled and there are no local and remote invalid tracks on the target side.
- aa. Target device is Read Write Enabled and there are no local and remote invalid tracks on the target side.
- ab. Not allowed on a diskless device.
- ac. RA is Ready on the target side.
- ad. SA or RA is Write Disabled or is Not Ready on the target side.
- ae. Must use -immediate.
- af. No local invalid tracks on the target side and no remote invalid tracks on the source side and must use -force.
- ag. Write Disabled on the SRDF link and must use -force.
- ah. Write Disabled on the SRDF link.
- ai. Source must be reachable.
- aj. Must be in async mode.
- ak. RA is Write Disabled on the target side.
- al. SA is Ready on the source side.
- am. SRDF/A must be active.
- an. Not allowed if the R1 or R2 array is running Enginuity 5977 or higher
- ao. Only allowed on the R1 side and must use -symforce.
- ap. Source must be reachable.

## 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

The following table 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**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 control operations and applicable pair states

| R1 -> R21 control operation: | R21 -> R2 pair state: |              |          |           |             |              |              |            |              |         |              |              |
|------------------------------|-----------------------|--------------|----------|-----------|-------------|--------------|--------------|------------|--------------|---------|--------------|--------------|
|                              | Sync in prog          | Synchronized | Split    | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid | Consistent   | TransmitIdle |
| createpair -establish        | ya,b,c,h,l,d,e        | ya,b,d,e     | ya,m,d,e | ya,m,d,e  | ya,m,k,e    | ya,m,d,e     |              |            |              |         | ya,b,c,h,l,e | ya,b,c,h,l,e |
| createpair -restore          | ya,f,h,l,e            | ya,e         | ya,m,d,e | ya,m,d,e  | ya,m,k,e    | ya,g,m,e     |              |            |              |         | ya,g,h,l,e   | ya,f,h,l,e   |
| createpair -invalidate R1    | ya,f,h,e              | ya,g,e       | ya,e     | ya,e      | ya,e        | ya,e         |              | ya,e       | ya,e         |         | ya,h,e       | ya,f,h,e     |
| createpair -invalidate R2    | ya,g,h,e              | ya,g,e       | ya,g,e   | ya,g,e    | ya,g,e      | ya,g,e       |              | ya,g,e     | ya,g,e       |         | ya,h,e       | ya,h,e       |
| deletepair                   | ya,m                  | ya,m         | ya       | ya        | ya          | ya           | ya,m         | ya,m       | ya,m         | ya,i    | ya,m         | ya,m         |
| half_deletepair              | ya,m                  | ya,m         | ya       | ya        | ya          | ya           | ya,m         | ya,m       | ya,m         | ya,i    | ya,m         | ya,m         |
| movepair                     | ya                    | ya           | ya       | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya           | ya           |
| half_movepair                | ya                    | ya           | ya       | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya           | ya           |
| swap                         | ya,m                  | ya,m         | ya       | ya        | ya          | ya           |              | ya,m       | ya,m         |         | ya,m         | ya,m         |
| half_swap                    | ya,m                  | ya,m         | ya       | ya        | ya          | ya           | ya,m         | ya,m       | ya,m         |         | ya,m         | ya,m         |
| swap -refresh R1             | ya,m                  | ya,m         | ya       | ya        | ya          | ya           |              | ya,m       | ya,m         |         | ya,m         | ya,m         |
| swap -refresh R2             |                       |              | ya       | ya        | ya,k        | ya           |              |            |              |         |              |              |
| establish                    | yc,l                  | yj           | ym       | ym        | ym,k        | ym           |              |            |              |         | yj,c,l       | yj,b,c,l     |
| establish -full              | yb,c,l                | yb           | ym       | ym        | ym,k        | ym           |              |            |              |         | yj,b,c,l     | yj,b,c,l     |
| split                        | Y                     | Y            | Y        | Y         | Y           | Y            | Y            | Y          | Y            | Y       | Y            | Y            |



**Table 46** R1 -> R21 cascaded SRDF control operations and applicable pair states (continued)

| R1 -> R21 control operation: | R21 -> R2 pair state: |              |       |           |             |              |              |            |              |         |            |              |
|------------------------------|-----------------------|--------------|-------|-----------|-------------|--------------|--------------|------------|--------------|---------|------------|--------------|
|                              | Sync in prog          | Synchronized | Split | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid | Consistent | TransmitIdle |
| restore                      | Yf,l                  | Y            | Ym    | Ym        | Ym,k        | Yg,m         |              |            |              |         | Y,l        | Yf,g,l       |
| restore -full                | Yf,l                  | Y            |       | Ym        | Ym,k        | Yg,m         |              |            |              |         | Y,l        | Yf,g,l       |
| update                       | Yf,l                  | Y            | Ym    | Ym        | Ym          | Ym           |              | Y          | Y            |         | Yl         | Yf,l         |
| failback                     | Yf,l                  | Y            | Ym    | Ym        | Ym,k        | Yg,m         |              |            |              |         | Y          | Yf,g,l       |
| failover                     | Y                     | Y            | Y     | Y         | Y           | Y            | Y            |            |              | Y       | Y          | Y            |
| failover -establish          | Ya,m                  | Ya,m         | Ya,m  | Ya,m      | Ya,m,k      | Ya,m         |              |            |              |         | Ya,m       | Ya,m         |
| failover -restore            | Ya, n,o,m             | Ya,o,m       | Ya,m  | Ya,m      | Ya,m,k      | Ya,m         | Ya,o,m       |            |              | Yo,m    | Ya,n,o,m   | Ya,n,o,m     |
| 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                        | Yp                    | Yp           | Yp    | Yp        | Yp          | Yp           |              | Yp         | Yp           |         | Yp         | Yp           |
| 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                       | Yc,f,l                | Yf           | Ym    | Ym        | Ym,k        | Ym           |              |            |              |         | Yj,c,l     | Yj,b,c,f,l   |
| 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            |

**Table 46** R1 -> R21 cascaded SRDF control operations and applicable pair states (continued)

| R1 -> R21 control operation:  | R21 -> R2 pair state: |              |       |           |             |              |              |            |              |         |            |              |
|-------------------------------|-----------------------|--------------|-------|-----------|-------------|--------------|--------------|------------|--------------|---------|------------|--------------|
|                               | Sync in prog          | Synchronized | Split | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid | Consistent | TransmitIdle |
| 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            |

- a. 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.
- b. Must use -force.
- c. Not allowed if operation results in R1->R21<-R2 data resynchronization.
- d. Not allowed when what will become the R21->R2 is in Synchronous mode.
- e. 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.
- f. Not allowed if R21 is diskless and operation will result in R1<-R21->R2 data resynchronization.
- g. If tracks are owed to R21 while R21->R2 is in the Transmit Idle state, data resynchronization between R1->R21 cannot complete.
- h. 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 Engenuity level lower than 5876 Q42012 SR.
- i. Not allowed if R1 is diskless and the SRDF link of the other pair is RW.
- j. If the other pair (not being controlled) is enabled for SRDF consistency protection, must use -force.
- k. The R21 is not visible to any host.
- l. 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.
- m. Not allowed if R21 is diskless.
- n. Not allowed if R2 owes tracks to R21.
- o. Must use -remote.
- p. Not allowed if R21 is diskless and both mirrors of R21 have invalid tracks.

## Cascaded SRDF: R21 - R2 control operations allowed for R1 - R21 pair states

The following table 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**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 SRDF control operations and applicable pair states

| R21 -> R2 control operation | R1 -> R21 pair state |              |                |              |                |              |              |              |              |         |              |              |
|-----------------------------|----------------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|--------------|---------|--------------|--------------|
|                             | Sync in prog         | Synchronized | Split          | Suspended    | Failed over    | Partitioned1 | Partitioned2 | R1 updated   | R1 updinprog | Invalid | Consistent   | TransmitIdle |
| createpair -establish       | γa,b,c               | γa,b,c       | γa,d,c         | γa,d,c       | γa,d,c         | γa,d,c       |              | γa,c         | γa,c         |         | γa,c         | γa,c         |
| createpair -restore         |                      |              | γa,e,c         | γa,c         | γa,e,c         | γa,e,c       |              | γa,c         | γa,c         |         |              |              |
| createpair -invalidate R1   | γa,c                 | γa,c         | γa,c           | γa,c         | γa,e,c         | γa,e,c       |              | γa,c         | γa,c         |         | γa,c         | γa,c         |
| createpair -invalidate R2   | γa,c                 | γa,c         | γa,d,c         | γa,d,c       | γa,d,c         | γa,d,c       |              | γa,c         | γa,c         |         | γa,e,c       | γa,c         |
| deletepair                  | γa,f                 | γa,f         | γa             | γa           | γa             | γa           | γa,f         | γa,f         | γa,f         | γa      | γa,f         | γa,f         |
| half_deletepair             | γa,f                 | γa,f         | γa             | γa           | γa             | γa           | γa,f         | γa,f         | γa,f         | γa      | γa,f         | γa,f         |
| movepair                    | γa                   | γa           | γa             | γa           | γa             | γa           | γa           | γa           | γa           | γa      | γa           | γa           |
| half_movepair               | γa                   | γa           | γa             | γa           | γa             | γa           | γa           | γa           | γa           | γa      | γa           | γa           |
| swap                        | γa,g,f,<br>c         | γa,g,f,<br>c | γa,g,c         | γa,g,c       | γa,g,c         | γa,h,g,<br>c |              | γa,g,f,<br>c | γa,h,g,<br>c |         | γa,g,f,<br>c | γa,g,f,<br>c |
| half_swap                   | γa,g,f,<br>c         | γa,g,f,<br>c | γa,g,c         | γa,g,c       | γa,g,c         | γa,g,c       | γa,g,f,<br>c | γa,g,f,<br>c | γa,g,f,<br>c |         | γa,g,f,<br>c | γa,g,f,<br>c |
| swap -refresh R1            | γa,g,f,<br>c         | γa,g,f,<br>c | γa,g,c         | γa,g,c       | γa,g,c         | γa,g,c       |              | γa,g,f,<br>c | γa,g,f,<br>c |         | γa,g,f,<br>c | γa,g,f,<br>c |
| swap -refresh R2            | γa,g,f,<br>c         | γa,g,f,<br>c | γa,g,c         | γa,g,c       | γa,g,c         | γa,g,c       |              | γa,g,f,<br>c | γa,g,f,<br>c |         | γa,g,f,<br>c | γa,g,f,<br>c |
| establish                   | γb                   | γb           | γd             | γd           | γd             | γd           |              | Y            | Y            |         | Y            | Y            |
| full establish              | γb                   | γb           | γd             | γd           | γd             | γd           |              | Y            | Y            |         | Y            |              |
| split                       | γf                   | γf           | Y              | Y            | Y              | Y            | γf           | γf           | γf           | γi      | 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              | γf           |              | Y            | Y            |         |              |              |
| failover                    |                      |              | Y              | Y            | Y              | Y            | Y            | γf           | γf           | γi      |              |              |
| failover -establish         |                      |              | γa,g,f,<br>c   | γa,g,f,<br>c | γa,g,f,j<br>,c | γa,g,c       |              |              |              |         |              |              |
| failover -restore           |                      |              | γa,d,g,<br>f,c | γa,g,f,<br>c | γa,g,f,j<br>,c |              |              |              |              |         |              |              |
| invalidate -R1              | Y                    | Y            | Y              | Y            | Y              | Y            |              | Y            | Y            |         | Y            | Y            |
| invalidate -R2              | Y                    | Y            | Y              | Y            | Y              | Y            |              | Y            | Y            |         | Y            | Y            |

**Table 47** R21 -> R2 cascaded SRDF control operations and applicable pair states (continued)

| R21 -> R2 control operation   | R1 -> R21 pair state |                  |                |                |                  |                  |                |                  |                  |                |                  |                  |
|-------------------------------|----------------------|------------------|----------------|----------------|------------------|------------------|----------------|------------------|------------------|----------------|------------------|------------------|
|                               | Sync in prog         | Synchronized     | Split          | Suspended      | Failed over      | Partitioned1     | Partitioned2   | R1 updated       | R1 updinprog     | Invalid        | Consistent       | TransmitIdle     |
| merge                         | Y <sup>d,k</sup>     | Y <sup>d,k</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d,k</sup> | Y <sup>d,k</sup> |                | Y <sup>d,k</sup> | Y <sup>d,k</sup> |                | Y <sup>d,k</sup> | Y <sup>d,k</sup> |
| 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 <sup>i</sup>   | Y                |                |                  |                  |                |                  |                  |
| refresh R2                    | Y                    | Y                | Y              | Y              | Y                | Y                |                | Y                | Y                |                | Y                | Y                |
| suspend                       | Y <sup>f</sup>       | Y <sup>f</sup>   | Y              | Y              | Y                | Y                | Y <sup>f</sup> | Y <sup>f</sup>   | Y <sup>f</sup>   | Y <sup>i</sup> | Y                | Y                |
| resume                        | Y <sup>l,b</sup>     | Y <sup>l,b</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup>   | Y <sup>d</sup>   |                | Y                | Y                |                | Y <sup>l</sup>   | Y <sup>e,l</sup> |
| 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_devspace       | Y <sup>m</sup>       | Y <sup>m</sup>   | Y              | Y              | Y                | Y                |                | Y <sup>m</sup>   | Y <sup>m</sup>   |                | Y                | Y                |
| deactivate -rdfa_devspace     | Y                    | Y                | Y              | Y              | Y                | Y                | Y              | Y                | Y                | Y              | Y                | Y                |
| activate -rdfa_pace           | Y <sup>m</sup>       | Y <sup>m</sup>   | Y              | Y              | Y                | Y                |                | Y <sup>m</sup>   | Y <sup>m</sup>   |                | Y                | Y                |
| deactivate -rdfa_pace         | Y                    | Y                | Y              | Y              | Y                | Y                | Y              | Y                | Y                | Y              | Y                | Y                |
| activate -rdfa_wpace          | Y <sup>m</sup>       | Y <sup>m</sup>   | Y              | Y              | Y                | Y                |                | Y <sup>m</sup>   | Y <sup>m</sup>   |                | 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                |

- a. 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.

**Table 47** R21 -> R2 cascaded SRDF control operations and applicable pair states (continued)

- b. 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.
- c. 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.
- d. Not allowed if R21 is diskless and operation will result in R1 <-R21-> R2 data resynchronization.
- e. If tracks are owed to R21 while R21->R2 is in the Transmit Idle state, data resynchronization between R1->R21 cannot complete.
- f. Not allowed if R21 is diskless.
- g. Not allowed if operation creates a concurrent R22 device on a array running on an Enginuity level lower than 5773.150.
- h. If the other pair (not being controlled) is enabled for SRDF consistency protection, must use -force.
- i. Not allowed if R1 is diskless and the SRDF link of the other pair is RW.
- j. The R21 is not visible to any host.
- k. Not allowed if operation results in local invalid tracks on the R21 device.
- l. Not allowed if operation results in data flowing from R2 -> R21.
- m. If the R1->R21 pair is operating in adaptive copy mode and is read/write (RW) on the SRDF link, must use -force.

## 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.

### Note

Devices that are part of an SRDF/Metro configuration cannot also be part of a cascaded configuration. There are no columns for ActiveActive or ActiveBias pair states in the tables in this section.

### Cascaded SRDF: R1 - R21 set operations allowed for R21 - R2 pair states

The following table 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**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** R1 → R21 cascaded SRDF set operations and applicable pair states

| R1 → R21 set operation:     | R21 → R2 pair state |              |       |           |             |              |              |            |               |         |                |                |
|-----------------------------|---------------------|--------------|-------|-----------|-------------|--------------|--------------|------------|---------------|---------|----------------|----------------|
|                             | Sync in prog        | Synchronized | Split | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 up/in prog | Invalid | Consistent     | TransitIdle    |
| set mode async <sup>a</sup> | 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 <sup>b</sup>      | Y            | Y     | Y         | Y           | Y            | Y            | Y          | Y             | Y       | Y <sup>b</sup> | Y <sup>b</sup> |
| set mode acp_wp             | Y <sup>b</sup>      | Y            | Y     | Y         | Y           | Y            | Y            | Y          | Y             | Y       | Y <sup>b</sup> | Y <sup>b</sup> |

- Async mode is not supported on both sides of an R21. It is only supported on one side or the other.
- 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.

### Cascaded SRDF: R21 - R2 set operations allowed for R1 - R21 pair states

The following table 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**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** R21 -> R2 Cascaded RDF Set Operations and Applicable Pair States

| R21 -> R2 set operation: | R1 -> R21 pair state |                |       |           |             |              |                |                |                |                |            |              |
|--------------------------|----------------------|----------------|-------|-----------|-------------|--------------|----------------|----------------|----------------|----------------|------------|--------------|
|                          | Sync in prog         | Synchronized   | Split | Suspended | Failed over | Partitioned1 | Partitioned2   | R1 updated     | R1 updinprog   | Invalid        | Consistent | TransmitIdle |
| set mode async           | Y <sup>a</sup>       | Y <sup>a</sup> | Y     | Y         | Y           | Y            | Y <sup>b</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>b</sup> | 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            |

a. Must use -force if all of these conditions apply:

1. 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.
2. The R21->R2 SRDF device pair (that is being controlled) is read/write (RW) on the SRDF link.
3. 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.

b. 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 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), including:

- [Concurrent SRDF: 1st leg R1- 2nd leg R1 pair states on page 432](#)
- [Concurrent SRDF: 1st leg R2 - 2nd leg R2 pair states on page 435](#)

---

### Note

Devices that are part of an SRDF/Metro configuration cannot be R11 or R22 devices. There are no columns for ActiveActive or ActiveBias pair states in the tables in this section.

---

## 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: 1st leg R1- 2nd leg R1 pair states

The following table 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**Partitioned2** pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

**Table 50** SRDF control operations and applicable states for concurrent R1 pairs

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R1 pair: |                  |                |                |                |                |                  |                |                  |                  |                    |                    |
|----------------------------------------------------------|---------------------------------------------------|------------------|----------------|----------------|----------------|----------------|------------------|----------------|------------------|------------------|--------------------|--------------------|
|                                                          | Sync in prog                                      | Synchronized     | Split          | Suspended      | Failed over    | Partitioned1   | Partitioned2     | R1 updated     | R1 updinprog     | Invalid          | Consistent         | Transmitidle       |
| createpair -establish                                    | Y <sup>a</sup>                                    | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup>   | Y <sup>a</sup>     | Y <sup>a</sup>     |
| createpair -restore                                      | Y <sup>a,b,c</sup>                                | Y <sup>a,c</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a,c</sup> |                |                  | Y <sup>a,b</sup> | Y <sup>a,b,c</sup> | Y <sup>a,b,c</sup> |
| createpair -invalidate R1                                | Y <sup>a,b</sup>                                  | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a,b</sup> | Y <sup>a</sup>   | Y <sup>a,b</sup>   | Y <sup>a,b</sup>   |
| createpair -invalidate R2                                | Y <sup>a</sup>                                    | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup>   | Y <sup>a</sup>     | Y <sup>a</sup>     |



**Table 50** SRDF control operations and applicable states for concurrent R1 pairs (continued)

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R1 pair: |                    |                    |                    |                      |                    |                    |                    |                      |                    |                      |                          |
|----------------------------------------------------------|---------------------------------------------------|--------------------|--------------------|--------------------|----------------------|--------------------|--------------------|--------------------|----------------------|--------------------|----------------------|--------------------------|
|                                                          | Sync in prog                                      | Synchronized       | Split              | Suspended          | Failed over          | Partitioned1       | Partitioned2       | R1 updated         | R1 updinprog         | Invalid            | Consistent           | Transmitidle             |
| deletepair                                               | γ <sup>a</sup>                                    | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>           |
| half_deletepair                                          | γ <sup>a</sup>                                    | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>           |
| movepair                                                 | γ <sup>a</sup>                                    | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>           |
| half_movepair                                            | γ <sup>a</sup>                                    | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>     | γ <sup>a</sup>       | γ <sup>a</sup>           |
| swap                                                     | γ <sup>a,d,e,f</sup>                              | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,d,f</sup>   | γ <sup>a,d,f</sup>       |
| half_swap                                                | γ <sup>a,d,e,f</sup>                              | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   |                    | γ <sup>a,d,f</sup>   | γ <sup>a,f</sup>         |
| swap -refresh R1                                         | γ <sup>a,b,d,e,f</sup>                            | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,b,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,b,d,f</sup> | γ <sup>a,b,d,f</sup>     |
| swap -refresh R2                                         | γ <sup>a,d,e,f</sup>                              | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup> | γ <sup>a,e,f</sup>   | γ <sup>a,e,f</sup> | γ <sup>a,d,f</sup>   | γ <sup>a,d,f</sup>       |
| 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                  | γ <sup>g</sup>       | Y                  | Y                  | γ <sup>h</sup>     |                      | Y                  | Y                    | Y                        |
| restore                                                  | γ <sup>b,c</sup>                                  | γ <sup>c</sup>     | Y                  | Y                  | Y                    | Y                  | γ <sup>c</sup>     |                    |                      | γ <sup>c</sup>     | γ <sup>b,c</sup>     | γ <sup>b,c</sup>         |
| full restore                                             | γ <sup>b,c</sup>                                  | γ <sup>c</sup>     | Y                  | Y                  | Y                    | Y                  | γ <sup>c</sup>     |                    |                      | γ <sup>c</sup>     | γ <sup>b,c</sup>     | γ <sup>b,c</sup>         |
| update                                                   | γ <sup>b,c</sup>                                  | γ <sup>c</sup>     | Y                  | Y                  | Y                    | Y                  | Y                  |                    |                      | γ <sup>c</sup>     | γ <sup>b,c</sup>     | γ <sup>b,c</sup>         |
| failback                                                 | γ <sup>b</sup>                                    | Y                  | γ <sup>b</sup>     | γ <sup>b</sup>     | γ <sup>b</sup>       | γ <sup>b</sup>     | γ <sup>b</sup>     | Y                  | γ <sup>b</sup>       | γ <sup>c</sup>     | γ <sup>b</sup>       | γ <sup>b</sup>           |
| failover                                                 | Y                                                 | Y                  | Y                  | Y                  | Y                    | Y                  | Y                  | Y                  | Y                    | Y                  | Y                    | Y                        |
| failover -establish                                      | γ <sup>a,d,i,f</sup>                              | γ <sup>a,f</sup>   | γ <sup>a,j,f</sup> | γ <sup>a,j,f</sup> | γ <sup>a,j,k,f</sup> | γ <sup>a,j,f</sup> | γ <sup>a,j,f</sup> |                    |                      |                    | γ <sup>a,d,i,f</sup> | γ <sup>a,d,i,f</sup>     |
| failover -restore                                        | γ <sup>a,l,d,i,f</sup>                            | γ <sup>a,f</sup>   | γ <sup>a,j,f</sup> | γ <sup>a,j,f</sup> | γ <sup>a,j,k,f</sup> | γ <sup>a,j,f</sup> | γ <sup>a,j,f</sup> |                    |                      |                    | γ <sup>a,d,i,f</sup> | γ <sup>a,m,l,d,i,f</sup> |
| invalidate -R1                                           | 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                        |
| merge                                                    | γ <sup>b</sup>                                    | Y                  | Y                  | Y                  | Y                    | Y                  | Y                  | Y                  | γ <sup>b</sup>       |                    | γ <sup>b</sup>       | γ <sup>b</sup>           |
| 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                        |

**Table 50** SRDF control operations and applicable states for concurrent R1 pairs (continued)

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R1 pair: |                |       |           |             |              |                |            |                |                  |                  |                  |
|----------------------------------------------------------|---------------------------------------------------|----------------|-------|-----------|-------------|--------------|----------------|------------|----------------|------------------|------------------|------------------|
|                                                          | Sync in prog                                      | Synchronized   | Split | Suspended | Failed over | Partitioned1 | Partitioned2   | R1 updated | R1 updinprog   | Invalid          | Consistent       | Transmitidle     |
| 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 <sup>b</sup>                                    | Y              | Y     | Y         | Y           | Y            | Y              | Y          | Y <sup>b</sup> |                  | Y <sup>b</sup>   | Y <sup>b</sup>   |
| 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 <sup>b,n</sup>                                  | Y <sup>n</sup> | Y     | Y         | Y           | Y            | Y <sup>c</sup> | Y          | Y <sup>b</sup> | Y <sup>b,c</sup> | Y <sup>b,n</sup> | Y <sup>b,n</sup> |
| 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                |

- 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.
- Not allowed if R2→R1←R2 data resynchronization will result.
- Must use -remote.
- 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.
- Not allowed when what will become the R21→R2 is in Synchronous mode.
- Not allowed if what will become the R21, R1 or R2 array is running HYPERMAX OS/Solutions Enabler 8.0.1 or higher.
- Must use -force. The state of the other pair changes to Suspended.

**Table 50** SRDF control operations and applicable states for concurrent R1 pairs (continued)

- h. Changes the state of the other pair to Split.
- i. 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.
- j. Not allowed if R11 is diskless.
- k. The R11 is not visible to any host.
- l. Not allowed if a diskless device and will become an R21 device and results in R1<-R21->R2 data resynchronization.
- m. If device will become an R21 and the other pair is in Transmit Idle pair state, data synchronization between R1->R21 cannot complete.
- n. Only allowed if data flows from R2 to R11 and -remote is used.

## Concurrent SRDF: 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.

The following table lists the allowable control operations for the R21->R2 pair given the SRDF pair states for the R1->R21 pair. 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.

**Partitioned1** pair state indicates that the remote array is in the SYMAPI database and was discovered.

**Partitioned2** pair state indicates the remote array is not in the SYMAPI database and was not discovered, or was removed from this database.

**Table 51** SRDF control operations and applicable states for concurrent R2 pairs

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R2 pair: |              |        |           |             |              |              |            |              |         |            |              |
|----------------------------------------------------------|---------------------------------------------------|--------------|--------|-----------|-------------|--------------|--------------|------------|--------------|---------|------------|--------------|
|                                                          | Sync in prog                                      | Synchronized | Split  | Suspended | Failed over | Partitioned1 | Partitioned2 | R1 updated | R1 updinprog | Invalid | Consistent | Transmitidle |
| createpair -establish                                    |                                                   |              | ya,b,c | ya,c      | ya,b,c      | ya,c         |              |            |              |         |            |              |
| createpair -restore                                      |                                                   |              | ya,b,c | ya,c      | ya,b,c      | ya,c         |              |            |              |         |            |              |
| createpair -invalidate R1                                | ya,c                                              | ya,c         | ya,b,c | ya,c      | ya,b,c      | ya,c         |              | ya,c       | ya,c         |         | ya,c       | ya,c         |
| createpair -invalidate R2                                | ya,c                                              | ya,c         | ya,b,c | ya,c      | ya,b,c      | ya,c         |              | ya,c       | ya,c         |         | ya,c       | ya,c         |
| deletepair                                               | ya                                                | ya           | ya     | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya         | ya           |
| half_deletepair                                          | ya                                                | ya           | ya     | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya         | ya           |
| movepair                                                 | ya                                                | ya           | ya     | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya         | ya           |
| half_movepair                                            | ya                                                | ya           | ya     | ya        | ya          | ya           | ya           | ya         | ya           | ya      | ya         | ya           |
| swap                                                     | ya,c                                              | ya,c         | ya,c   | ya,c      | ya,c        | ya,c         |              | ya,c       | ya,c         |         | ya,c       | ya,c         |
| half_swap                                                | ya,c                                              | ya,c         | ya,c   | ya,c      | ya,c        | ya,c         |              | ya,c       | ya,c         |         | ya,c       | ya,c         |

**Table 51** SRDF control operations and applicable states for concurrent R2 pairs (continued)

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R2 pair: |                  |                  |                    |                    |                  |              |                    |                    |         |                  |                  |
|----------------------------------------------------------|---------------------------------------------------|------------------|------------------|--------------------|--------------------|------------------|--------------|--------------------|--------------------|---------|------------------|------------------|
|                                                          | Sync in prog                                      | Synchronized     | Split            | Suspended          | Failed over        | Partitioned1     | Partitioned2 | R1 updated         | R1 updinprog       | Invalid | Consistent       | TransmitIdle     |
| swap -refresh R1                                         | Y <sup>a,c</sup>                                  | Y <sup>a,c</sup> | Y <sup>a,c</sup> | Y <sup>a,c</sup>   | Y <sup>a,c</sup>   | Y <sup>a,c</sup> |              | Y <sup>a,c</sup>   | Y <sup>a,c</sup>   |         | Y <sup>a,c</sup> | Y <sup>a,c</sup> |
| swap -refresh R2                                         |                                                   |                  |                  | Y <sup>a,c</sup>   |                    | Y <sup>a,c</sup> |              |                    |                    |         |                  |                  |
| establish                                                |                                                   |                  | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| establish -full                                          |                                                   |                  | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| split                                                    |                                                   |                  | Y                | Y <sup>d</sup>     | Y                  | Y                |              | Y                  | Y                  | Y       |                  |                  |
| restore                                                  |                                                   |                  | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| restore -full                                            |                                                   |                  | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| update                                                   |                                                   |                  | Y                | Y                  | Y                  | Y                |              |                    |                    |         |                  |                  |
| failback                                                 |                                                   |                  | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| failover                                                 |                                                   |                  | Y                | Y <sup>d</sup>     | Y                  | Y                | Y            |                    |                    | Y       |                  |                  |
| failover -establish                                      |                                                   |                  | Y <sup>c</sup>   | Y <sup>d,c</sup>   | Y <sup>e,c</sup>   | Y                |              |                    |                    |         |                  |                  |
| failover -restore                                        |                                                   |                  | Y <sup>a,c</sup> | Y <sup>a,d,c</sup> | Y <sup>a,e,c</sup> | Y <sup>a,c</sup> |              | Y <sup>a,f,c</sup> | Y <sup>a,f,c</sup> |         |                  |                  |
| 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                |              |                    |                    |         |                  |                  |
| 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 <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                | Y            | Y <sup>b</sup>     | Y <sup>b</sup>     | 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 <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              |                    |                    |         |                  |                  |
| rw_disable R2                                            | Y                                                 | Y                | Y <sup>b</sup>   | Y                  | Y <sup>b</sup>     | Y                |              | Y <sup>b</sup>     | Y <sup>b</sup>     |         | Y                | Y                |
| rw_enable R1                                             | Y                                                 | Y                | Y                | Y                  | Y                  | Y                |              | Y                  | Y                  | Y       | Y                | Y                |

**Table 51** SRDF control operations and applicable states for concurrent R2 pairs (continued)

| Control operation of 1st leg of concurrent SRDF R1 pair: | Pair state of 2nd leg of concurrent SRDF R2 pair: |              |                |                |                |              |              |                |                |         |            |              |
|----------------------------------------------------------|---------------------------------------------------|--------------|----------------|----------------|----------------|--------------|--------------|----------------|----------------|---------|------------|--------------|
|                                                          | Sync in prog                                      | Synchronized | Split          | Suspended      | Failed over    | Partitioned1 | Partitioned2 | R1 updated     | R1 updinprog   | Invalid | Consistent | TransmitIdle |
| rw_enable R2                                             |                                                   |              | Y              | Y <sup>d</sup> | 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 <sup>b</sup> | Y              | Y <sup>b</sup> | Y            |              | Y <sup>b</sup> | Y <sup>b</sup> |         | 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            |

- a. 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.
- b. Must use -force. The state of the other pair changes to Suspended.
- c. Not allowed if what will become the R21, R1 or R2 array is running HYPERMAX OS/Solutions Enabler 8.0.1 or higher.
- d. Changes the state of the other pair to Split.
- e. The other pair's (the one not being controlled) R1 is not visible to any host.
- f. 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.

## Consistency group 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 52** SRDF control operations and applicable pair states for devices in an SRDF/CG

| Control operation: | Pair state:    |              |       |           |             |                |                |            |              |         |            |
|--------------------|----------------|--------------|-------|-----------|-------------|----------------|----------------|------------|--------------|---------|------------|
|                    | Sync in prog   | Synchronized | Split | Suspended | Failed over | Partitioned1   | Partitioned2   | R1 updated | R1 updinprog | Invalid | Consistent |
| enable             | Y              | Y            | Y     | Y         | Y           |                |                | Y          | Y            | Y       | Y          |
| disable            | Y              | Y            | Y     | Y         | Y           | Y <sup>b</sup> | Y <sup>b</sup> | Y          | Y            | Y       | Y          |
| modify -add        | Y <sup>a</sup> | Y            |       |           |             |                |                |            |              |         | Y          |
| modify -remove     | Y <sup>a</sup> | Y            |       |           |             |                |                |            |              |         | Y          |
| modify -recover    | Y              | Y            |       | Y         |             | Y <sup>b</sup> | Y <sup>b</sup> |            |              |         | Y          |

- a. There are no local invalid tracks on the source side and no remote invalid tracks on the remote side.
- b. Must use -force.

# APPENDIX B

## SRDF operations and TimeFinder sessions

This appendix describes the following topics:

- [SRDF operations during TimeFinder/Snap/VP Snap sessions.....](#) 440
- [SRDF set operations for TimeFinder/Snap sessions.....](#) 450
- [SRDF operations for TimeFinder/Clone sessions.....](#) 453
- [SRDF set operations for TimeFinder/Clone sessions.....](#) 463
- [SRDF operations for Extent-level TimeFinder/Clone sessions.....](#) 466
- [SRDF set operations for Extent-level TimeFinder/Clone sessions.....](#) 476
- [SRDF operations for TimeFinder Snapvx.....](#) 478
- [SRDF set operations for TimeFinder Snapvx sessions.....](#) 487

## 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 cannot be paced. For additional details, see [Devices that cannot be paced in a cascaded SRDF configuration on page 150](#).

## SRDF operations when R1 is source of TimeFinder Snap/VP Snap

The following table identifies the allowable SRDF actions when the R1 is the source of a TimeFinder/Snap or VP Snap session.

**Table 53** Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Copy on write  | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|----------------|-----------------|----------|-------------------|---------|--------|
| createpair -establish     | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| createpair -restore       | Y          | Y              | Y              | Y              | Y <sup>b</sup> | Y <sup>b</sup>  | Y        | Y                 |         |        |
| createpair -invalidate R1 | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| createpair -invalidate R2 | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | 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 <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| half_swap                 | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| swap -refresh R1          | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| swap -refresh R2          | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| establish                 | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| establish -full           | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |



**Table 53** Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation: | No session | Create in prog | Created        | Recreated      | Copy on write  | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|----------------|----------------|----------------|-----------------|----------|-------------------|---------|--------|
| split                   | Y          | Y              | Y              | Y              | Y              | Y               | Y        | Y                 | Y       | Y      |
| restore                 | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| restore -full           | Y          | Y              | Y              | Y              | Y <sup>c</sup> |                 | Y        | Y                 |         |        |
| update                  | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| failback                | Y          | Y              | Y              | Y              | Y              | Y <sup>a</sup>  | Y        | Y                 |         |        |
| failover                | Y          | Y              | Y              | Y              | Y              | Y               | Y        | Y                 | Y       | Y      |
| failover -establish     | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| failover -restore       | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |          | Y <sup>c</sup>    |         |        |
| 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 <sup>a</sup>  | 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 <sup>a</sup>  | 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 <sup>a</sup>  | 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      |

**Table 53** Allowable SRDF operations when R1 is the source of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Copy on write | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------------|-----------------|----------|-------------------|---------|--------|
| 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 -rdfawpace_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      |

- Only allowed for TimeFinder VP Snap.
- Allowed for TimeFinder VP Snap. If not TimeFinder VP Snap, must use -force.
- 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:
  - The SRDF pair is an R22→R1 of a concurrent R2 setup in which either:
    - The R22 array is running an Enginuity level less than 5876 or the R1 array is running an Enginuity level less than 5875..
    - SRDF/A device-level write pacing is not configured for autostart on the R2 side.
  - The SRDF pair is not an R22→R1 of a concurrent R2 setup and either:
    - The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - SRDF/A device-level write pacing is not configured for autostart on the R2 side.

## SRDF operations when R1 is target of TimeFinder Snap/VP Snap

The following table identifies the allowable SRDF actions when R1 is the target of a TimeFinder/Snap or VP Snap session.

**Table 54** Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap

| SRDF control operation: | No session | Create in prog | Created | Recreated | Copy on write | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------------|-----------------|----------|-------------------|---------|--------|
| createpair -restore     | Y          |                |         |           |               |                 | Y        | Y                 |         |        |

**Table 54** Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Copy on write | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------------|-----------------|----------|-------------------|---------|--------|
| 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      |
| 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      |

**Table 54** Allowable SRDF operations when R1 is the target of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Copy on write | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------------|-----------------|----------|-------------------|---------|--------|
| 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      |
| 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

The following table identifies the allowable SRDF actions when the R2 is the source of a TimeFinder/Snap or VP Snap copy session.

**Table 55** Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Copy on write  | Restore in prog | Restored       | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|----------------|-----------------|----------------|-------------------|---------|--------|
| createpair -establish     | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |                | Y <sup>c</sup>    |         |        |
| createpair -restore       | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |                | Y <sup>c</sup>    |         |        |
| createpair -invalidate R1 | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |                | Y <sup>c</sup>    |         |        |
| createpair -invalidate R2 | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 |                | Y <sup>c</sup>    |         |        |
| 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 <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y               | Y              | Y <sup>a</sup>    |         |        |
| half_movepair             | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y               | Y              | Y <sup>a</sup>    |         |        |
| 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 <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| establish -full           | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| split                     | Y          | Y              | Y              | Y              | Y              | Y               | Y              | Y                 | Y       | Y      |
| restore                   | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>b</sup>  | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| restore -full             | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>b</sup>  | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| update                    | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| failback                  | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| 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                 |         |        |

**Table 55** Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:     | No session | Create in prog | Created        | Recreated      | Copy on write  | Restore in prog | Restored       | Terminate in prog | Invalid | Failed |
|-----------------------------|------------|----------------|----------------|----------------|----------------|-----------------|----------------|-------------------|---------|--------|
| 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                 |         |        |
| suspend                     | Y          | Y              | Y              | Y              | Y              | Y               | Y              | Y                 | Y       | Y      |
| resume                      | Y          | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> | Y <sup>c</sup> |                 | Y <sup>c</sup> | Y <sup>c</sup>    |         |        |
| 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup>  | Y <sup>d</sup> | Y <sup>d</sup>    |         |        |
| activate -rdfa_pace         | Y          | Y              | Y              | Y              | Y              | Y               | Y              | Y                 | Y       | Y      |
| deactivate -rdfa_pace       | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup>  | Y <sup>d</sup> | Y <sup>d</sup>    |         |        |
| 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      |

**Table 55** Allowable SRDF operations when R2 is the source of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Copy on write | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------------|-----------------|----------|-------------------|---------|--------|
| deactivate -rdfa_wpace_exempt | Y          | Y              | Y       | Y         | Y             | Y               | Y        | Y                 | Y       | Y      |

- a. 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:
  1. The SRDF pair is an R21→R2 of a cascaded setup in which either:
    - a. The R21 array is running an Enginuity level lower than 5876, or the R2 array is running an Enginuity level lower than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
  2. The SRDF pair is not an R21→R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
- b. Only allowed if the required Enginuity levels or patches are detected. If TimeFinder VP Snap, requires Enginuity version 5876 Q42012 SR and higher.
- c. 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:
  1. The SRDF pair is an R21→R2 of a cascaded setup in which either:
    - a. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
  2. The SRDF pair is not an R21→R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
- d. If the SRDF/A session is in the Transmit Idle state, you must issue the command with -symforce from the R1 side.

## SRDF operations when R2 is target of TimeFinder Snap/VP Snap

The following table identifies the allowable SRDF actions when the R2 is the target of a TimeFinder/Snap or VP Snap copy session.

**Table 56** Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap

| SRDF control operation: | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------------|--------|-----------------|----------|-------------------|---------|--------|
| createpair -establish   | Y          |                |         |           |               |        |                 |          | Y                 |         |        |

**Table 56** Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------------|--------|-----------------|----------|-------------------|---------|--------|
| 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                 |         |        |
| 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      |



**Table 56** Allowable SRDF operations when R2 is the target of a TimeFinder/Snap or VP Snap (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------------|--------|-----------------|----------|-------------------|---------|--------|
| 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      |
| 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 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

The following table identifies the allowable SRDF set actions when the R1 is the source of a TimeFinder/Snap copy session.

**Table 57** Allowable SRDF set operations when R1 is the source of a TimeFinder/Snap

| SRDF control operation: | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | 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

The following table identifies the allowable SRDF set actions when the R1 is the target for a TimeFinder/Snap copy session.

**Table 58** Allowable SRDF set operations when R1 is the target of a TimeFinder/Snap

| SRDF control operation: | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | 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

The following table identifies the allowable SRDF set actions when the R2 is the source of a TimeFinder/Snap copy session.

**Table 59** Allowable SRDF set operations when R2 is the source of a TimeFinder/Snap

| SRDF set operation: | No session | Create in prog | Created        | Recreated      | Copy on write  | Restore in prog | Restored       | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|----------------|----------------|----------------|-----------------|----------------|-------------------|---------|--------|
| set mode async      | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> |                 | Y <sup>a</sup> | Y <sup>a</sup>    |         |        |
| 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      |

- a. 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:
  1. The RDF device pair is the R21→R2 of a cascaded setup and either:
    - a. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 group of the R21 device.
  2. The SRDF pair is not the R21→R2 of a cascaded setup and either:
    - a. The R1 or the R2 array is running an Enginuity level lower than 5875.
    - b. 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

The following table identifies the allowable SRDF set when the R2 is the target of a TimeFinder/Snap copy session.

**Table 60** Allowable SRDF set operations when R2 is the target of a TimeFinder/Snap

| SRDF set operation: | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | 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      |

**Table 60** Allowable SRDF set operations when R2 is the target of a TimeFinder/Snap (continued)

| SRDF set operation: | No session | Create in prog | Created | Recreated | Copy on write | Copied | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------------|--------|-----------------|----------|-------------------|---------|--------|
| 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 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

The following table identifies the allowable SRDF actions when the R1 is the source of a TimeFinder/Clone copy session.

**Table 61** Allowable SRDF operations when R1 is the source of a TimeFinder/Clone

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| half_swap                 | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| swap -refresh R1          | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| swap -refresh R2          | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| 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                 |         |        |

**Table 61** Allowable SRDF operations when R1 is the source of a TimeFinder/Clone (continued)

| SRDF control operation:  | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|--------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| failover                 | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y              | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| failover -establish      | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| failover -restore        | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| 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      |

**Table 61** Allowable SRDF operations when R1 is the source of a TimeFinder/Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |

- a. 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:
1. The SRDF pair is an R22->R1 of a concurrent R2 setup in which either:
    - a. The R22 array is running an Enginuity level less than 5876 or the R1 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R2 side.
  2. The SRDF device pair is not an R22->R1 of a concurrent R2 setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R2 side.

## SRDF operations when R1 is target of TimeFinder Clone

The following table identifies the allowable SRDF actions when the R1 is the target of a TimeFinder/Clone copy session.

**Table 62** Allowable SRDF operations when R1 is the target of a TimeFinder/Clone

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |

**Table 62** Allowable SRDF operations when R1 is the target of a TimeFinder/Clone (continued)

| SRDF control operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog   | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------|----------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 <sup>a</sup> | Y      |               |                | Y     |                 | Y        | Y                 |         |        |
| full establish          | Y          |                |         |           |         | Y <sup>a</sup> | 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                 |         |        |
| 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                 |         |        |



**Table 62** Allowable SRDF operations when R1 is the target of a TimeFinder/Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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_devspace       | Y          | Y              | Y       | Y         | Y       | Y            | Y      | Y             | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| deactivate -rdfa_devspace     | 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      |

a. Not allowed if the devices are in Active RDF mode.

---

#### Note

There are no subscripts for TimeFinder/Clone R1 targets.

---

## SRDF operations when R2 is source of TimeFinder Clone

The following table identifies the allowable SRDF actions when the R2 is the source of a TimeFinder/Clone copy session.

**Table 63** Allowable SRDF operations when R2 is the source of a TimeFinder/Clone

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write    | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|------------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| createpair -establish     | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>a,d</sup> | Y <sup>d</sup> | Y     |                 |          | Y <sup>d</sup>    |         |        |
| createpair -restore       | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 |          | Y <sup>d</sup>    |         |        |
| createpair -invalidate R1 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 |          | Y                 |         |        |
| createpair -invalidate R2 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | 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 <sup>b</sup> | Y <sup>b</sup> | Y <sup>b</sup> | Y       | Y            | Y      | Y <sup>b</sup>   | Y <sup>b</sup> | Y     | Y               | Y        | Y <sup>b</sup>    |         |        |
| half_movepair             | Y          | Y <sup>b</sup> | Y <sup>b</sup> | Y <sup>b</sup> | Y       | Y            | Y      | Y <sup>b</sup>   | Y <sup>b</sup> | Y     | Y               | Y        | Y <sup>b</sup>    |         |        |
| 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| establish -full           | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>a,d</sup> | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| split                     | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y                | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| restore                   | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     | Y <sup>c</sup>  | Y        | Y <sup>d</sup>    |         |        |
| restore -full             | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     | Y <sup>c</sup>  | Y        | Y <sup>d</sup>    |         |        |
| update                    | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| failback                  | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| 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      |
| 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      |

**Table 63** Allowable SRDF operations when R2 is the source of a TimeFinder/Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     | Y <sup>d</sup>  | Y        | Y <sup>d</sup>    |         |        |
| activate -rdfa_pace           | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y              | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| deactivate -rdfa_pace         | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     | Y <sup>d</sup>  | Y        | Y <sup>d</sup>    |         |        |
| 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      |

- a. Only allowed if required Enginuity levels or patches are detected and not a TimeFinder VP Snap.  
b. 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:

**Table 63** Allowable SRDF operations when R2 is the source of a TimeFinder/Clone (continued)

1. The SRDF pair is an R21->R2 of a cascaded setup in which either:
  - a. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
  - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
2. The SRDF pair is not an R21->R2 of a cascaded setup and either:
  - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
  - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
- c. Not allowed if the devices are in async mode and there is a TimeFinder/Clone off of the R2 either:
  1. The SRDF pair will become an R21->R2 for which any of the following apply:
    - a. 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.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
    - c. If the R21->R2 pair will be read/write (RW) on the SRDF link, the R21 must be pace-capable.
  2. The SRDF device pair is not an R21->R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
- d. 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:
  1. The SRDF pair is an R21->R2 of a cascaded setup in which either:
    - a. The R21 Symmetrix is running an Enginuity level less than 5876 or the R2 Symmetrix is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
  2. The SRDF pair is not an R21->R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 Symmetrix array.
    - b. ii. SRDF/A device-level write pacing is not configured for autostart on the R1 side.

## SRDF operations when R2 is target of TimeFinder Clone

The following table identifies the allowable SRDF actions when the R2 is the target of a TimeFinder/Clone copy session.

**Table 64** Allowable SRDF operations when R2 is the target of a TimeFinder/Clone

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| createpair -establish     | Y          |                |         |           |         |              |        |               |                |       |                 |          | Y                 |         |        |
| createpair -restore       | Y          |                |         |           |         |              |        |               |                |       |                 |          | Y                 |         |        |
| createpair -invalidate R1 | Y          |                |         |           |         |              |        |               |                |       |                 |          | Y                 |         |        |
| createpair -invalidate R2 | Y          |                |         |           |         |              |        |               |                |       |                 |          | Y                 |         |        |

**Table 64** Allowable SRDF operations when R2 is the target of a TimeFinder/Clone (continued)

| SRDF control operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |
| 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                 |         |        |

**Table 64** Allowable SRDF operations when R2 is the target of a TimeFinder/Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 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

The following table identifies the allowable SRDF set actions when the R1 is the source of a TimeFinder/Clone copy session.

**Table 65** Allowable SRDF set operations when R1 is the source of a TimeFinder/Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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

The following table identifies the allowable SRDF set actions when the R1 is the target of a TimeFinder/Clone copy session.

**Table 66** Allowable SRDF set operations when R1 is the target of a TimeFinder/Clone

| SRDF set operation: | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|----------------|----------------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| set mode async      | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | 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      |

a. Not allowed if TimeFinder/Clone pair was created with -copy.

## SRDF set operations when R2 is source of TimeFinder/Clone

The following table identifies the allowable SRDF set actions when the R2 is the source of a TimeFinder/Clone copy session.

**Table 67** Allowable SRDF set operations when R2 is the source of a TimeFinder/Clone

| SRDF set operation: | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| set mode async      | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 | Y        | Y <sup>a</sup>    |         |        |
| 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      |

- a. 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:
  1. The RDF device pair is the R21→R2 of a cascaded setup and either:
    - a. The R21 array is running an Enginuity level lower than 5876 or the R2 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 group of the R21 device.
  2. The SRDF pair is not the R21→R2 of a cascaded setup and either:
    - a. The R1 or the R2 array is running an Enginuity level lower than 5875.
    - b. 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

The following table identifies the allowable SRDF set actions when the R2 is the target of a TimeFinder/Clone copy session.

**Table 68** Allowable SRDF set operations when R2 is the target of a TimeFinder/Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 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

The following table identifies the allowable SRDF actions when the R1 is the source of an Extent-level TimeFinder/Clone copy session.

**Table 69** Allowable SRDF operations when R1 is the source of an Extent-level Clone

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |
| _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 <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| half_swap                 | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| swap -refresh R1          | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| swap -refresh R2          | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| 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                 |         |        |

**Table 69** Allowable SRDF operations when R1 is the source of an Extent-level Clone (continued)

| SRDF control operation:  | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|--------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| failover                 | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y              | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| failover -establish      | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| failover -restore        | Y          | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y       | Y            | Y      | Y <sup>a</sup> | Y <sup>a</sup> | Y     |                 |          | Y <sup>a</sup>    |         |        |
| 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      |

**Table 69** Allowable SRDF operations when R1 is the source of an Extent-level Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |

- a. 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:
1. The SRDF pair is an R22->R1 of a concurrent R2 setup in which either:
    - a. The R22 array is running an Enginuity level lower than 5876, or the R1 array is running an Enginuity level lower than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R2 side.
  2. The SRDF device pair is not an R22->R1 of a concurrent R2 setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. 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

The following table identifies the allowable SRDF actions when the R1 is the target of an Extent-level TimeFinder/Clone copy session.

**Table 70** Allowable SRDF operations when the R1 is the target of an Extent-level Clone

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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          |                |         |           |         |              |        |               |                |       |                 |          |                   |         |        |

**Table 70** Allowable SRDF operations when the R1 is the target of an Extent-level Clone (continued)

| SRDF control operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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          |                |         |           |         |              |        |               |                |       |                 |          |                   |         |        |
| 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      |

**Table 70** Allowable SRDF operations when the R1 is the target of an Extent-level Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 Extent-level TimeFinder/Clone R1 targets.

## SRDF operations when R2 is source of Extent-level Clone

The following table identifies the allowable SRDF actions when the R2 is the source of an Extent-level TimeFinder/Clone copy session.

**Table 71** Allowable SRDF operations when R2 is the source of an Extent-level Clone

| SRDF control operation:   | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write    | Copy on access | Split          | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|------------------|----------------|----------------|-----------------|----------|-------------------|---------|--------|
| createpair -establish     | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> |         | Y            | Y      | Y <sup>a,d</sup> | Y <sup>d</sup> | Y              |                 |          | Y <sup>d</sup>    |         |        |
| createpair -restore       | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y              |                 |          | Y <sup>d</sup>    |         |        |
| createpair -invalidate R1 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y              |                 |          | Y                 |         |        |
| createpair -invalidate R2 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y <sup>d</sup> |                 |          | Y                 |         |        |

**Table 71** Allowable SRDF operations when R2 is the source of an Extent-level Clone (continued)

| SRDF control operation: | No session | Create in prog | Created        | Recreated      | Precopy        | Copy in prog | Copied | Copy on write    | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|----------------|----------------|----------------|--------------|--------|------------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 <sup>b</sup> | Y <sup>b</sup> | Y <sup>b</sup> | Y              | Y            | Y      | Y <sup>b</sup>   | Y <sup>b</sup> | Y     | Y               | Y        | Y <sup>b</sup>    |         |        |
| half_movepair           | Y          | Y <sup>b</sup> | Y <sup>b</sup> | Y <sup>b</sup> | Y              | Y            | Y      | Y <sup>b</sup>   | Y <sup>b</sup> | Y     | Y               | Y        | Y <sup>b</sup>    |         |        |
| 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y              | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| establish -full         | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y              | Y            | Y      | Y <sup>a,d</sup> | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| split                   | Y          | Y              | Y              | Y              | Y              | Y            | Y      | Y                | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| restore                 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y              | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     | Y <sup>c</sup>  | Y        | Y <sup>d</sup>    |         |        |
| restore -full           | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y              | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     | Y <sup>c</sup>  | Y        | Y <sup>d</sup>    |         |        |
| update                  | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y              | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| failback                | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y            | Y      | Y <sup>d</sup>   | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| 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      |
| 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                 |         |        |

**Table 71** Allowable SRDF operations when R2 is the source of an Extent-level Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created        | Recreated      | Precopy | Copy in prog | Copied | Copy on write  | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|----------------|----------------|---------|--------------|--------|----------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| suspend                       | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y              | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| resume                        | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     |                 | Y        | Y <sup>d</sup>    |         |        |
| rw_disable R2                 | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     | Y               | Y        | Y <sup>d</sup>    | 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 <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     | Y <sup>d</sup>  | Y        | Y <sup>d</sup>    |         |        |
| activate -rdfa_pace           | Y          | Y              | Y              | Y              | Y       | Y            | Y      | Y              | Y              | Y     | Y               | Y        | Y                 | Y       | Y      |
| deactivate -rdfa_pace         | Y          | Y <sup>d</sup> | Y <sup>d</sup> | Y <sup>d</sup> | Y       | Y            | Y      | Y <sup>d</sup> | Y <sup>d</sup> | Y     | Y <sup>d</sup>  | Y        | Y <sup>d</sup>    |         |        |
| 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      |

- a. Only allowed if required Enginuity levels or patches are detected.
- b. 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:
  1. The SRDF pair is an R21->R2 of a cascaded setup in which either:
    - a. The R21 array is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
  2. The SRDF pair is not an R21->R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side of the new group.
- c. Not allowed if the devices are in async mode and there is a TimeFinder/Clone off of the R2 and either:
  1. The SRDF pair will become an R21->R2 for which any of the following apply:



**Table 71** Allowable SRDF operations when R2 is the source of an Extent-level Clone (continued)

- a. 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.
- b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
- c. If the R21→R2 pair will be read/write (RW) on the SRDF link, the R21 must be pace-capable.
- 2. The SRDF device pair is not an R21→R2 of a cascaded setup and either:
  - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
  - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.
- d. 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:
  - 1. The SRDF pair is an R21→R2 of a cascaded setup in which either:
    - a. The R21 Symmetrix is running an Enginuity level less than 5876 or the R2 array is running an Enginuity level less than 5875.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R21 side.
  - 2. The SRDF pair is not an R21→R2 of a cascaded setup and either:
    - a. The Enginuity level is lower than 5875 on either the R1 or R2 array.
    - b. SRDF/A device-level write pacing is not configured for autostart on the R1 side.

## SRDF operations when R2 is target of Extent-level Clone

The following table identifies the allowable SRDF actions when the R2 is the target of an Extent-level TimeFinder/Clone copy session.

**Table 72** Allowable SRDF operations when the R2 is the target of an Extent-level Clone

| SRDF control operation:   | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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          |                |         |           |         |              |        |               |                |       |                 |          |                   |         |        |

**Table 72** Allowable SRDF operations when the R2 is the target of an Extent-level Clone (continued)

| SRDF control operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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      |
| 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      |

**Table 72** Allowable SRDF operations when the R2 is the target of an Extent-level Clone (continued)

| SRDF control operation:       | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|-------------------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 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

The following table identifies the allowable SRDF set actions when the R1 is the source of an Extent-level TimeFinder/Clone copy session.

**Table 73** Allowable SRDF set operations when R1 is the source of an Extent-level Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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

The following table identifies the allowable SRDF set actions when the R1 is the target of an Extent-level TimeFinder/Clone copy session.

**Table 74** Allowable SRDF set operations when R1 is the target of an Extent-level Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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

The following table identifies the allowable SRDF set actions when the R2 is the source of an Extent-level TimeFinder/Clone copy session.

**Table 75** Allowable SRDF set operations when the R2 is the source of an Extent-level Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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

The following table identifies the allowable SRDF set actions when the R2 is the target of an Extent-level TimeFinder/Clone copy session.

**Table 76** Allowable SRDF set operations when R2 is the target of an Extent-level Clone

| SRDF set operation: | No session | Create in prog | Created | Recreated | Precopy | Copy in prog | Copied | Copy on write | Copy on access | Split | Restore in prog | Restored | Terminate in prog | Invalid | Failed |
|---------------------|------------|----------------|---------|-----------|---------|--------------|--------|---------------|----------------|-------|-----------------|----------|-------------------|---------|--------|
| 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 Snapvx

This section lists the allowable SRDF operations for TimeFinder Snapvx 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 77** SnapVX State Determination

| Snap VX State         | Snap VX CLI                       | Determine State                                                                                               |
|-----------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------|
| Established           | symsnapvx list                    | The flags field indicates:<br>(F)ailed: = . for No Failure                                                    |
| Restore In Progress   | symsnapvx list -restored -detail  | The Done(%) field is not at 100%.                                                                             |
| Restored              | symsnapvx 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:<br>(F)ailed = X for Failed                                                         |
| Link Copy In Progress | symsnapvx list -link -tgt -detail | The flags field indicates:<br>(C)opy = I for CopyInProg<br>The Done (%) field is not at 100%.                 |
| Link Copied           | symsnapvx list -link -tgt -detail | The flags field indicates:<br>(C)opy = C for Copied or D for Copied/Destaged<br>The Done (%) field is at 100% |
| Linked                | symsnapvx list -linked            | The flags field indicates :<br>(C)opy = . for NoCopy Link                                                     |

### SRDF operations when R1 is source of Snapvx

The following table identifies allowable SRDF operations when the R1 is the source of a TimeFinder Snapvx session

**Table 78** Allowable SRDF operations when R1 is the source of a TimeFinder Snapvx

| SRDF control operation:    | No snapshot | Establish in progress | Established | Restore in prog | Restored | Terminate in prog | 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                 |        |
| 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 R1 is the source of a TimeFinder Snapvx (continued)

| SRDF control operation:       | No snapshot | Establish in progress | Established | Restore in prog | Restored | Terminate in prog | 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                 |        |
| 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

The following table identifies allowable SRDF operations when the R1 is the target of a TimeFinder Snapvx session



**Table 79** Allowable SRDF operations when R1 is the target of a TimeFinder Snapvpx

| 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 <sup>a</sup>        | Y           |        |        |
| establish -full            | Y       | Y <sup>a</sup>        | 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      |
| not_ready R1               | Y       | Y                     | Y           |        | Y      |
| not_ready R2               | Y       | Y                     | Y           |        | Y      |

**Table 79** Allowable SRDF operations when R1 is the target of a TimeFinder Snapvx (continued)

| SRDF control operation:       | No Link | Link copy in progress | Link copied | Linked | Failed |
|-------------------------------|---------|-----------------------|-------------|--------|--------|
| 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      |

a. Not allowed if the devices are in Active RDF mode.

## SRDF operations when R2 is source of Snapvx

The following table identifies allowable SRDF operations when the R2 is the source of a TimeFinder Snapvx session

**Table 80** Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx

| SRDF control operation:    | No snapshot | Establish in progress | Established    | Restore in prog | Restored | Terminate in prog | Failed |
|----------------------------|-------------|-----------------------|----------------|-----------------|----------|-------------------|--------|
| createpair - establish     | Y           | Y                     | Y <sup>a</sup> |                 |          | 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 <sup>a</sup> |                 | 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 80** Allowable SRDF operations when R2 is the source of a TimeFinder Snapvx (continued)

| SRDF control operation:       | No snapshot | Establish in progress | Established    | Restore in prog | Restored       | Terminate in prog | 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 <sup>b</sup>        | Y <sup>b</sup> | Y <sup>b</sup>  | Y <sup>b</sup> | Y <sup>b</sup>    |        |
| activate -rdfa_pace           | Y           | Y                     | Y              | Y               | Y              | Y                 | Y      |
| deactivate -rdfa_pace         | Y           | Y <sup>b</sup>        | Y <sup>b</sup> | Y <sup>b</sup>  | Y <sup>b</sup> | Y <sup>b</sup>    |        |
| 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      |

a. Must use -force

b. If the SRDF/A session is in Transmit Idle state, you must use symforce from the R1 side.

## SRDF operations when R2 is target for Snapvx

The following table identifies allowable SRDF operations when the R2 is the target of a TimeFinder Snapvx session.

**Table 81** Allowable SRDF operations when R2 is the target of a TimeFinder Snapvpx

| 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      |
| merge                      | Y       |                       | Y           |        |        |
| msc_cleanup                | Y       | Y                     | Y           |        | Y      |
| not_ready R1               | Y       | Y                     | Y           |        | Y      |
| not_ready R2               | Y       | Y                     | Y           |        | Y      |

**Table 81** Allowable SRDF operations when R2 is the target of a TimeFinder Snapvx (continued)

| SRDF control operation:       | No Link | Link copy in progress | Link copied | Linked | Failed |
|-------------------------------|---------|-----------------------|-------------|--------|--------|
| 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 Snapvpx sessions

This section lists the allowable SRDF set operations for Timefinder Snapvpx sessions on the R1 and R2 source, and the R1 and R2 target.

### SRDF set operations when R1 is source of TimeFinder Snapvpx

The following table identifies allowable SRDF set operations when the R1 is the source of a TimeFinder Snapvpx session.

**Table 82** Allowable SRDF set operations when R1 is the source of a TimeFinder Snapvpx

| SRDF set operation: | No snapshot | Establish in progress | Established | Restore in prog | Restored | Terminate in prog | 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 Snapvpx

The following table identifies allowable SRDF set operations when the R1 is the target of a TimeFinder Snapvpx session.

**Table 83** Allowable SRDF set operations when R1 is the target of a TimeFinder Snapvpx

| 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

The following table identifies allowable SRDF set operations when the R2 is the source of a TimeFinder Snapvx session.

**Table 84** Allowable SRDF set operations when R2 is the source of a TimeFinder Snapvx

| SRDF set operation: | No snapshot | Establish in progress | Established | Restore in prog | Restored | Terminate in prog | 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

The following table identifies allowable SRDF set operations when the R2 is the target of a TimeFinder Snapvx session.

**Table 85** 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:

- [rcopy session on the R1 side](#).....490
- [rcopy session on the R2 side](#).....494

## rcopy session on the R1 side

This section lists the allowable SRDF operations and the applicable rcopy states when there is an rcopy session on the R1.

### Allowable SRDF operations when R1 is part of an rcopy PUSH

The following table identifies the allowable SRDF operations and applicable rcopy states when there is an rcopy PUSH session on the R2.

Allowed SRDF operations are noted by Ys.

**Table 86** Allowed SRDF operations when the R1 is part of an rcopy PUSH

| SRDF control operation:   | rcopy state:   |                |                  |                |                |                  |                      |                  |                   |                |                |                    |                    |                    |                |                |                |                |
|---------------------------|----------------|----------------|------------------|----------------|----------------|------------------|----------------------|------------------|-------------------|----------------|----------------|--------------------|--------------------|--------------------|----------------|----------------|----------------|----------------|
|                           | None           | Create in prog | Created          | Copy in prog   | Copy on write  | Copied           | Recreate in progress | Recreated        | Terminate in prog | Failed         | Invalid        | Verify in progress | Restored           | Restore in prog    | Precopy        | Sync in prog   | Synchronized   | Stopped        |
| createpair -establish     | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>   | Y <sup>a</sup>       | Y <sup>a</sup>   | Y <sup>a</sup>    | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>     | Y <sup>a</sup>     | Y <sup>b,c,a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> |
| createpair -restore       | Y              |                | Y <sup>b,a</sup> |                |                | Y <sup>b,a</sup> |                      | Y <sup>b,a</sup> |                   |                |                |                    | Y <sup>b,c,a</sup> |                    |                |                |                |                |
| createpair -invalidate R1 | Y              |                | Y <sup>b,a</sup> |                |                | Y <sup>b,a</sup> |                      | Y <sup>b,a</sup> |                   |                |                |                    | Y <sup>b,c,a</sup> |                    |                |                |                |                |
| createpair -invalidate R2 | Y              | Y              | Y                | Y              | Y              | Y                | Y                    | Y                | Y                 | Y              | Y              | Y                  | Y <sup>b,c,a</sup> | 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 <sup>d</sup>     | Y <sup>b,c</sup>   | Y              | Y              | Y              | Y              |
| establish -full           | Y              | Y              | Y                | Y              | Y              | Y                | Y                    | Y                | Y                 | Y              | Y              | Y                  | Y <sup>e</sup>     | Y <sup>b,c</sup>   | Y              | Y              | Y              | Y              |
| split                     | Y              | Y              | Y                | Y              | Y              | Y                | Y                    | Y                | Y                 | Y              | Y              | Y                  | Y <sup>e</sup>     | Y                  | Y              | Y              | Y              | Y              |
| restore                   | Y              |                | Y <sup>b</sup>   |                |                | Y <sup>b</sup>   |                      | Y <sup>b</sup>   |                   |                |                |                    | Y <sup>b,c</sup>   |                    |                |                |                |                |
| restore -full             | Y              |                | Y <sup>b</sup>   |                |                | Y <sup>b</sup>   |                      | Y <sup>b</sup>   |                   |                |                |                    | Y <sup>b,c</sup>   |                    |                |                |                |                |
| update R1                 | Y              |                | Y <sup>b</sup>   |                |                | Y <sup>b</sup>   |                      | Y <sup>b</sup>   |                   |                |                |                    | Y <sup>b,c</sup>   |                    |                |                |                |                |
| failback                  | Y              |                | Y <sup>b</sup>   |                |                | Y <sup>b</sup>   |                      | Y <sup>b</sup>   |                   |                |                |                    | Y <sup>b,c</sup>   |                    |                |                |                |                |

**Table 86** Allowed SRDF operations when the R1 is part of an rcopy PUSH (continued)

| SRDF control operation:   | rcopy state: |                |                |                |                |                |                      |                |                   |                |                  |                    |                  |                  |                |                |                |                |
|---------------------------|--------------|----------------|----------------|----------------|----------------|----------------|----------------------|----------------|-------------------|----------------|------------------|--------------------|------------------|------------------|----------------|----------------|----------------|----------------|
|                           | None         | Create in prog | Created        | Copy in prog   | Copy on write  | Copied         | Recreate in progress | Recreated      | Terminate in prog | Failed         | Invalid          | Verify in progress | Restored         | Restore in prog  | Precopy        | Sync in prog   | Synchronized   | 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 <sup>b</sup> |                |                | Y <sup>b</sup> |                      | Y <sup>b</sup> |                   |                |                  |                    | Y <sup>b,c</sup> |                  |                |                |                |                |
| invalidate -R2            | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y                | Y                  | Y <sup>c</sup>   | Y <sup>b,c</sup> | Y              | Y              | Y              | Y              |
| merge                     | Y            | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup>       | Y <sup>e</sup> | Y <sup>e</sup>    | Y <sup>e</sup> | Y <sup>e</sup>   | Y <sup>e</sup>     | Y <sup>e</sup>   | Y <sup>e</sup>   | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> |
| 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 <sup>b</sup> |                |                | Y <sup>b</sup> |                      |                |                   |                | Y <sup>b,c</sup> |                    |                  |                  |                |                |                |                |
| refresh R2                | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y <sup>c</sup>   | Y <sup>b,c</sup>   | Y                | Y                | 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 <sup>e</sup> | Y <sup>f</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>f</sup> | Y <sup>e</sup>       | Y <sup>f</sup> | Y <sup>e</sup>    | Y <sup>e</sup> | Y <sup>e,c</sup> | Y <sup>e,c</sup>   | Y <sup>f</sup>   | Y <sup>e</sup>   | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> | Y <sup>e</sup> |
| 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_devspace   | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y                | Y                  | Y                | Y                | Y              | Y              | Y              | Y              |
| deactivate -rdfa_devspace | 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              |

**Table 86** Allowed SRDF operations when the R1 is part of an rcopy PUSH (continued)

| SRDF control operation:           | rcopy state: |                |         |              |               |        |                      |           |                   |        |         |                    |          |                 |         |              |              |         |
|-----------------------------------|--------------|----------------|---------|--------------|---------------|--------|----------------------|-----------|-------------------|--------|---------|--------------------|----------|-----------------|---------|--------------|--------------|---------|
|                                   | None         | Create in prog | Created | Copy in prog | Copy on write | Copied | Recreate in progress | Recreated | Terminate in prog | Failed | Invalid | Verify in progress | Restored | Restore in prog | Precopy | Sync in prog | Synchronized | Stopped |
| activate -<br>rdfa_wpace_exempt   | Y            | Y              | Y       | Y            | Y             | Y      | Y                    | Y         | Y                 | Y      | Y       | Y                  | Y        | Y               | Y       | Y            | Y            | Y       |
| deactivate -<br>rdfa_wpace_exempt | Y            | Y              | Y       | Y            | Y             | Y      | Y                    | Y         | Y                 | Y      | Y       | Y                  | Y        | Y               | Y       | Y            | Y            | Y       |

- Not allowed if the R1 array is running HYPERMAX OS (5977) or above.
- Not allowed if the R1 array is running an Enginuity level lower than 5874.
- Not allowed if the Rcopy session has front end zero detect.
- Not allowed if the Rcopy session has front end zero detect.
- Not allowed if the R2 owes data to the R1.
- 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.

## Allowable SRDF operations when R1 is part of an rcopy PULL

The following table identifies the allowable SRDF operations and applicable rcopy states when there is an rcopy PULL session on the R1.

Allowed SRDF operations are noted by Ys.

**Table 87** Allowed SRDF operations when the R1 is part of an rcopy PULL

| SRDF control operation:   | rcopy state: |                |         |              |                |        |                   |        |         |                    |              |              |         |          |
|---------------------------|--------------|----------------|---------|--------------|----------------|--------|-------------------|--------|---------|--------------------|--------------|--------------|---------|----------|
|                           | None         | Create in prog | Created | Copy in prog | Copy on access | Copied | Terminate in prog | Failed | Invalid | Verify in progress | Sync in prog | Synchronized | 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        |

**Table 87** Allowed SRDF operations when the R1 is part of an rcopy PULL (continued)

| SRDF control operation: | rcopy state: |                |         |              |                |        |                   |        |         |                    |              |              |         |          |
|-------------------------|--------------|----------------|---------|--------------|----------------|--------|-------------------|--------|---------|--------------------|--------------|--------------|---------|----------|
|                         | None         | Create in prog | Created | Copy in prog | Copy on access | Copied | Terminate in prog | Failed | Invalid | Verify in progress | Sync in prog | Synchronized | Stopped | Failback |
| 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            | ya,b           | ya,b    | ya,b         | ya,b           | ya,b   | ya,b              | ya,b   | ya,b    | ya,b               | ya,b         | ya,b         | ya,b    | ya,b     |
| establish -full         | Y            | ya,b           | ya,b    | ya,b         | ya,b           | ya,b   | ya,b              | ya,b   | ya,b    | ya,b               | ya,b         | ya,b         | ya,b    | ya,b     |
| 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            | yb             | yb      | yb           | yb             | yb     | yb                | yb     | yb      | yb                 | yb           | yb           | yb      | yb       |
| 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       |          |
| 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            | ya,b,c         | ya,c    | ya,b         | ya,b,b         | ya,c   | ya,b,c            | ya,b,c | ya,b,c  | ya,b,c             | ya,b,c       | ya,b,c       | ya,b,c  | ya,b,c   |
| rw_disable R2           | Y            | Y              | Y       | Y            | Y              | Y      | Y                 | Y      | Y       | Y                  | Y            | Y            | Y       | Y        |

**Table 87** Allowed SRDF operations when the R1 is part of an rcopy PULL (continued)

| SRDF control operation:       | rcopy state: |                |         |              |                |        |                   |        |         |                    |              |              |         |          |
|-------------------------------|--------------|----------------|---------|--------------|----------------|--------|-------------------|--------|---------|--------------------|--------------|--------------|---------|----------|
|                               | None         | Create in prog | Created | Copy in prog | Copy on access | Copied | Terminate in prog | Failed | Invalid | Verify in progress | Sync in prog | Synchronized | Stopped | Failback |
| 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_devspace       | Y            | Y              | Y       | Y            | Y              | Y      | Y                 | Y      | Y       | Y                  | Y            | Y            | Y       | Y        |
| deactivate -rdfa_devspace     | 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       |          |
| 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        |

- a. Not allowed if the R1 array is running an Engenuity level lower than 5874.
- b. Not allowed if the R2 owes data to the R1.
- c. Not allowed if the Rcopy session has front end zero detect.

## rcopy session on the R2 side

This section lists the allowable SRDF operations and the applicable rcopy states when there is an rcopy session on the R2.

### Allowable SRDF operations when R2 is part of an rcopy PUSH

The following table identifies the allowable SRDF operations and applicable rcopy states when there is an rcopy PUSH session on the R2.

Allowed SRDF operations are noted by Ys.

**Table 88** Allowed SRDF operations when the R2 is part of an rcopy PUSH

| SRDF control operation:   | rcopy state: |                |                |                |                |                |                      |                |                   |                |                |                    |                         |                 |                |                |                |                |
|---------------------------|--------------|----------------|----------------|----------------|----------------|----------------|----------------------|----------------|-------------------|----------------|----------------|--------------------|-------------------------|-----------------|----------------|----------------|----------------|----------------|
|                           | None         | Create in prog | Created        | Copy in prog   | Copy on write  | Copied         | Recreate in progress | Recreated      | Terminate in prog | Failed         | Invalid        | Verify in progress | Restored                | Restore in prog | Precopy        | Sync in prog   | Synchronized   | 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 <sup>a</sup> |                |                | Y <sup>a</sup> |                      | Y <sup>a</sup> |                   |                |                |                    | Y <sup>a</sup> ,<br>b,c |                 |                |                |                |                |
| establish -full           | Y            |                | Y <sup>a</sup> |                |                | Y <sup>a</sup> |                      | Y <sup>a</sup> |                   |                |                |                    | Y <sup>a</sup> ,<br>b,c |                 |                |                |                |                |
| split                     | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y              | Y                  | Y                       | Y               | Y              | Y              | Y              | Y              |
| restore                   | Y            | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>       | Y <sup>a</sup> | Y <sup>a</sup>    | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>     | Y <sup>a</sup> ,c       | Y <sup>c</sup>  | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> |
| restore -full             | Y            | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>       | Y <sup>a</sup> | Y <sup>a</sup>    | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup>     | Y <sup>a</sup> ,c       | Y <sup>c</sup>  | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> | Y <sup>a</sup> |
| update                    | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y              | Y                  | Y <sup>c</sup>          | Y               | 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              |
| merge                     | Y            | Y              | Y              | Y              | Y              | Y              | Y                    | Y              | Y                 | Y              | Y              | Y                  | Y                       | Y               | Y              | Y              | Y              | Y              |

**Table 88** Allowed SRDF operations when the R2 is part of an rcopy PUSH (continued)

| SRDF control operation:       | rcopy state: |                |                |              |               |                |                      |                |                   |        |         |                    |                      |                 |         |              |              |         |
|-------------------------------|--------------|----------------|----------------|--------------|---------------|----------------|----------------------|----------------|-------------------|--------|---------|--------------------|----------------------|-----------------|---------|--------------|--------------|---------|
|                               | None         | Create in prog | Created        | Copy in prog | Copy on write | Copied         | Recreate in progress | Recreated      | Terminate in prog | Failed | Invalid | Verify in progress | Restored             | Restore in prog | Precopy | Sync in prog | Synchronized | Stopped |
| 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 <sup>a</sup> |              |               | Y <sup>a</sup> |                      | Y <sup>a</sup> |                   |        |         |                    | Y <sup>a, b, c</sup> |                 |         |              |              |         |
| 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_devspace       | Y            | Y              | Y              | Y            | Y             | Y              | Y                    | Y              | Y                 | Y      | Y       | Y                  | Y                    | Y               | Y       | Y            | Y            | Y       |
| deactivate -rdfa_devspace     | 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       |

- a. Not allowed if the R2 array is running an Engenuity level lower than 5874.
- b. Not allowed if donor update specified.
- c. Not allowed if the Rcopy session has front end zero detect.



## Allowable SRDF operations when R2 is part of an rcopy PULL

The following table identifies the allowable SRDF operations and applicable rcopy states when there is an rcopy PULL session on the R2.

Allowed SRDF operations are noted by Ys.

**Table 89** Allowed SRDF operations when the R2 is part of an rcopy PULL

| SRDF control operation:   | rcopy state: |                |         |              |                |        |                   |        |         |                    |              |              |         |          |
|---------------------------|--------------|----------------|---------|--------------|----------------|--------|-------------------|--------|---------|--------------------|--------------|--------------|---------|----------|
|                           | None         | Create in prog | Created | Copy in prog | Copy on access | Copied | Terminate in prog | Failed | Invalid | Verify in progress | Sync in prog | Synchronized | 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 89** Allowed SRDF operations when the R2 is part of an rcopy PULL (continued)

| SRDF control operation:       | rcopy state: |                |         |              |                |        |                   |        |         |                    |              |              |         |          |
|-------------------------------|--------------|----------------|---------|--------------|----------------|--------|-------------------|--------|---------|--------------------|--------------|--------------|---------|----------|
|                               | None         | Create in prog | Created | Copy in prog | Copy on access | Copied | Terminate in prog | Failed | Invalid | Verify in progress | Sync in prog | Synchronized | 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       |          |
| 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        |