

**EMC Solutions Enabler
Symmetrix SRDF Family CLI**
Version 6.0

PRODUCT GUIDE

**P/N 300-000-877
REV A06**

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.emc.com

Copyright © 2005 EMC® Corporation. All Rights Reserved.

Printed January, 2005

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, replication, or distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Preface	xiii
----------------------	------

Chapter 1 Overview

Introduction to SRDF	1-2
SRDF Site Configurations	1-3
SRDF Types of Implementation.....	1-5
SRDF Device Types and Groups.....	1-7
SRDF Devices	1-7
Dynamic RDF Devices	1-7
RDF Device Groups.....	1-7
Composite Groups.....	1-8
SRDF Consistency Groups	1-8
Command Summary	1-9

Chapter 2 SRDF Control Operations

Finding SRDF Devices.....	2-2
Device Groups.....	2-2
Composite Groups.....	2-2
Listing SRDF Devices.....	2-2
Query SRDF Devices	2-4
Ping SRDF Devices	2-4
Verify SRDF States	2-4
SRDF Operations.....	2-5
Types of SRDF Control Operations.....	2-5
Migrating Data from R1 to a Larger R2 Device.....	2-5
Composite SRDF Control Operations.....	2-6
Device External Locks.....	2-7
Disallow Synchronization Actions.....	2-8

Locking at the RA Group Level Instead of Symmetrix.....	2-8
SRDF Operations and Copy Sessions	2-8
Full Establish	2-9
Incremental Establish	2-11
Split	2-13
Full Restore	2-16
Incremental Restore	2-18
Failover	2-21
Failback	2-23
Update R1 Mirror	2-25
Create SRDF Pairs.....	2-28
Delete SRDF Pairs.....	2-28
Singular SRDF Control Operations.....	2-29
Invalid Tracks	2-30
Suspend.....	2-32
Resume	2-33
Read/Write Enable R1 Mirror	2-34
Read/Write Enable R2 Mirror	2-35
Write Disable R1 Mirror.....	2-35
Write Disable R2 Mirror.....	2-36
Read/Write Disable R2 Mirror	2-37
Refresh R1 Mirror	2-38
Refresh R2 Mirror	2-39
Invalidate R1 Mirror.....	2-40
Invalidate R2 Mirror.....	2-41
Ready R1 Mirror	2-41
Ready R2 Mirror	2-42
Not Ready R1 Mirror	2-42
Not Ready R2 Mirror	2-43
Merge Track Tables.....	2-44
Enable Consistency Protection	2-45
Disable Consistency Protection	2-46
Confirm R2 Data Copy	2-46
SRDF States.....	2-47
SRDF Pair States	2-47
RDF Operations and Applicable States	2-50
Setting SRDF Modes.....	2-54
Dynamic SRDF Pair Operations	2-59
Requirements	2-59
Display RDF Capable Devices	2-59
Create Device File	2-60
Creating Dynamic Pairs with a Device File	2-60
Creating Dynamic SRDF Pairs with Invalidate.....	2-61

Creating Dynamic SRDF Pairs with Establish	2-61
Creating Dynamic SRDF Pairs for a Restore	2-62
Createpair Restrictions	2-62
Creating Dynamic Concurrent SRDF Pairs	2-63
Delete Dynamic SRDF Pairs	2-64
Control of Dynamic Pairs by Device Group	2-66
Dynamic R1/R2 Swap	2-67
Display RDF Swap-Capable Devices	2-68
Swap RDF Devices	2-68
Dynamic Failover Establish	2-70
Restrictions	2-70
Command Functionality	2-71
symrdf -g Command Line Options	2-72
Targeting All Devices	2-75
Targeting BCV Devices	2-75
Bypassing Locks	2-76
Listing Devices	2-76
Running Repetitive Commands	2-77
Forcing a Rejected State	2-77
Forcing a Rejected State with Symforce	2-78
Getting Help	2-78
Setting No Echo Display	2-78
Setting No Prompt Confirmation	2-78
Obtaining Information from the SYMAPI Database	2-78
Remote Data Copying	2-78
Targeting a Symmetrix	2-79
Verifying Device States	2-79
Setting the Number of Invalid Track Updates	2-80
Displaying Command Status	2-80
Dropping the SRDF/A Session	2-80
symrdf -cg Command Line Options	2-81
symrdf -file Command Line Options	2-84

Chapter 3 Various Remote Operations

Multi-Hop Operations	3-2
Multi-Hop SRDF Sites	3-2
System-Wide Device Groups	3-2
System-Wide Splits	3-3
Targeting Commands to Various Multi-Hop Devices and Links	3-5
RDF Group Topologies in an SRDF	3-7
RDF Groups in a Point-to-Point SRDF Link	3-7

RDF Groups in a Switched SRDF Link.....	3-8
SRDF/Asynchronous Operations.....	3-9
SRDF/A Benefits and Features	3-9
SRDF/A Restrictions.....	3-11
Setting SRDF/Asynchronous Mode	3-12
SRDF/A Session Monitoring.....	3-12
SRDF/A Ordered-Write Processing.....	3-13
SRDF/A Session Status	3-15
Listing SRDF/A Device Information.....	3-15
Using the Immediate Option	3-17
Using BCVs to Preserve R2 SRDF/A Data Copy.....	3-18
Mode Transition to Synchronous	3-18
Dynamic SRDF Group Operations.....	3-19
Adding Dynamic Groups.....	3-19
Modifying Dynamic Groups.....	3-21
Removing Dynamic Groups	3-22
Concurrent RDF Operations	3-23
Supported Concurrent RDF Modes	3-24
Device Groups and RDF Groups.....	3-24
Viewing Concurrent RDF Devices	3-25
Establishing Concurrent RDF Devices	3-25
Splitting Concurrent RDF Devices.....	3-26
Restoring Concurrent RDF Devices.....	3-26
The Remote Option for Restore, Update, Failback	3-27
TimeFinder Consistent Splits Across RDF	3-28
Consistent Split on Both RDF Sides Using PowerPath.....	3-28
Enginuity Consistency Assist (ECA)	3-29
SRDF/Automated Replication Operations.....	3-31
Locked Devices	3-31
Single-Hop Data Copies	3-33
Multi-Hop Data Copies	3-37
Concurrent BCVs With SRDF/AR.....	3-40
Replication Cycle Patterns	3-43
Replication Log Entries.....	3-45
Clustered SRDF/AR Environments	3-45
Setting Replication Retry and Sleep Times.....	3-47
Symreplicate File Parameters	3-49
Relative Symmetrix Manager Functions	3-54
RDF Consistency Group Operations	3-55
PowerPath Consistency Protection.....	3-55
Multi Session Consistency (MSC) Protection for SRDF/A	3-57
RDF Daemon Support for MSC.....	3-57

Creating a Composite Group 3-61

Enable/Disable RDF Consistency 3-61

Deleting a Composite Group..... 3-62

Appendix A TimeFinder/Snap and Clone State Reference

Copy Session Pair States A-2

TimeFinder/Snap Pair States A-2

SRDF Operations for TimeFinder/Snap Copy Sessions ... A-3

TimeFinder Clone Pair States..... A-5

SRDF Operations for TimeFinder Clone Copy Sessions... A-5

Setting Snap and Clone Devices to Asynchronous Mode. A-8

Index i-1

Figures

1-1	SRDF Bidirectional Configuration	1-4
1-2	SRDF Campus Solution	1-5
1-3	Switched (Fabric) RDF Topology	1-6
2-1	Establishing an RDF Pair	2-10
2-2	Incremental Establish of an RDF Pair	2-12
2-3	Splitting an RDF Pair	2-14
2-4	Restoring an SRDF Device	2-17
2-5	Incrementally Restoring an SRDF Device	2-20
2-6	Failover of an SRDF Device	2-22
2-7	Failback of an SRDF Device	2-24
2-8	Update of SRDF Device Track Tables	2-26
2-9	RDF Pair and Link States	2-47
3-1	Various Remote Configurations	3-4
3-2	Remote Multi-Hop SRDF Configurations	3-6
3-3	RDF Group Topology in a Point-to-Point SRDF Solution	3-7
3-4	RDF Group Topology in a Switched RDF Solution	3-8
3-5	SRDF/Asynchronous Mode	3-13
3-6	Concurrent RDF	3-23
3-7	Restoring R1 in a Concurrent RDF	3-26
3-8	Restoring R1 and the Other R2 in a Concurrent RDF	3-27
3-9	Consistent Split on Both Sides	3-28
3-10	ECA Consistent Split	3-30
3-11	Automated Data Copy Path in Single-Hop SRDF Systems	3-33
3-12	Automated Data Copy Path in Multi-Hop SRDF Systems	3-38
3-13	Concurrent BCVs in a Single-Hop Configuration	3-40
3-14	Concurrent BCV in a Multi-Hop Configuration	3-43
3-15	All Data Propagation is Suspended Upon Any Link Failure	3-56
3-16	RDF Daemon Host Visibility	3-58

Tables

1-1	SRDF Command Summary	1-9
2-1	Composite SRDF Control Operations	2-6
2-2	Decomposition of Composite Operations Into Singular Operations ..	2-29
2-3	Singular SRDF Control Operations	2-31
2-4	SRDF Pair States	2-48
2-5	SRDF States for the RDF Devices and Link	2-49
2-6	RDF Control Operations and Applicable States	2-50
2-7	RDF Device Data Status for a Swap	2-69
2-8	RDF Device States Before Swap Operation	2-69
2-9	symrdf -g Control Arguments and Possible Options	2-72
2-10	symrdf -g View Arguments and Possible Options	2-73
2-11	symrdf -g View Arguments and Possible Options	2-74
2-12	symrdf -cg Control Arguments and Possible Options	2-81
2-13	symrdf -cg View Arguments and Possible Options	2-82
2-14	symrdf -file Control Arguments and Possible Options	2-84
2-15	symrdf -file View Arguments and Possible Options	2-86
3-1	Remote Multi-Hop SRDF Commands	3-5
3-2	Initial Setups for Cycle Timing Parameters	3-45
3-3	Symmetrix Manager Relationship	3-54
3-4	RDF Daemon Optional Behavior Parameters	3-60
A-1	TimeFinder/Snap Pair States	A-2
A-2	SRDF Operations for TimeFinder/Snap Copy Sessions	A-3
A-3	TimeFinder Clone Pair States	A-5
A-4	SRDF Operations for TimeFinder Clone Copy Sessions	A-6
A-5	Asynchronous for Snap and Clone Sessions	A-8

As part of its effort to continuously improve and enhance the performance and capabilities of the EMC product line, EMC periodically releases new versions of both the EMC Enginuity Operating Environment and EMC Solutions Enabler. Therefore, some functions described in this guide may not be supported by all versions of Enginuity or Solutions Enabler currently in use. For the most up-to-date information on product features, see your product release notes.

If an EMC Solutions Enabler feature does not function properly or does not function as described in this guide, please contact the EMC Customer Support Center for assistance.

Audience This manual provides both guide and reference information for command-line users and script programmers that describes how to manage devices in a Symmetrix Remote Data Facility (SRDF) using the SYMCLI commands of the EMC Solution Enabler software.

Organization The following defines the structure of this manual:

Chapter 1, *Overview*, highlights the SYMCLI SRDF Component features, SYMCLI commands, and terminology.

Chapter 2, *SRDF Control Operations*, describes how to monitor and control SRDF devices using SYMCLI.

Chapter 3, *Various Remote Operations*, describes the operation, management, and strategies of various possible SRDF configurations.

Appendix A, *TimeFinder/Snap and Clone State Reference*, describes the available SRDF actions for devices participating in a TimeFinder Clone or TimeFinder/Snap copy session.

Related Documentation

The following are EMC Solutions Enabler Symmetrix publications of related interest:

- ◆ *EMC Solutions Enabler Symmetrix CLI Command Reference*
- ◆ *EMC Solutions Enabler V6.0 Support Matrix*
- ◆ *EMC Solutions Enabler Installation Guide*
- ◆ *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*
- ◆ *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*
- ◆ *EMC Solutions Enabler Symmetrix Configuration Change CLI Product Guide*
- ◆ *EMC Host Connectivity Guides for [your operating system]*

The following are EMC Engineering white papers of related interest:

- ◆ *Using SYMCLI to Perform Control Operations with SRDF Family Products*
- ◆ *Using SYMCLI to Query and Verify with SRDF Family Products*
- ◆ *Using SYMCLI to Implement RDF Consistency Protection with SRDF Family Products*

Conventions Used in this Manual

The following conventions are used in this manual:

In this manual, every use of the word SYMCLI means EMC Solutions Enabler.

In this manual, every use of the word MVS means OS/390 and z/OS.

Note: A note calls attention to any item of information that may be of special importance to the reader.

**CAUTION**

A caution contains information essential to avoid damage or degraded integrity to storage of your data. The caution might also apply to protection of your software or hardware.

Typographical Conventions

This manual uses the following type style conventions in this guide:

bold text	Boldface text provides extra emphasis and emphasizes warnings, and specifies window names and menu items in text.
<i>italic text</i>	<i>Italic text and characters emphasizes new terms, identifies variables in a software syntax (nonliteral notation), identifies unique word usage, and applies emphasis in examples and in references to book titles and sections.</i>
fixed space courier font	A fixed space font identifies files and pathnames, and is used in command line entries, displayed text, or program listings.

Where to Get Help

EMC software products are supported directly by the EMC Customer Support Center.

Obtain technical support by calling the EMC Customer Support Center at one of the following numbers:

- United States:** (800) 782-4362 (SVC-4EMC)
- Canada:** (800) 543-4782 (543-4SVC)
- Worldwide:** (508) 497-7901

Language services are available upon request.

Sales and Customer Support Contacts

For the list of EMC sales locations, access the EMC home page at:

<http://www.EMC.com/contact/>

For additional information on the EMC products and services available to customers and partners, refer to the EMC Powerlink website at:

<http://powerlink.EMC.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please e-mail us at **techpub_comments@emc.com** to let us know about your opinion or any errors concerning this manual.

Your technical enhancement suggestions for future development consideration are also welcome. To send a suggestion, log on to <http://powerlink.emc.com>, follow the path **Support, Contact Support**, and choose **Software Product Enhancement Request** from the Subject menu.

Overview

This chapter provides an overview of the Symmetrix Remote Data Facility (SRDF) business continuance solution and describes the various SRDF site configurations used in a Symmetrix storage complex.

- ◆ Introduction to SRDF.....1-2
- ◆ SRDF Site Configurations1-3
- ◆ SRDF Device Types and Groups1-7
- ◆ Command Summary1-9

Introduction to SRDF

The Symmetrix Remote Data Facility (SRDF®) is a business continuance solution that maintains a mirror image of data at the device level in Symmetrix® arrays located in physically separate sites.

The Solutions Enabler SRDF component extends the basic SYMCLI command set to include SRDF commands that allow you to perform control operations on remotely located RDF devices. This manual specifically describes the functionality of:

- ◆ SRDF — General monitor and control operations
- ◆ SRDF/S — Synchronous mode
- ◆ SRDF/A — Asynchronous mode
- ◆ SRDF/AR — Automated Replication
- ◆ SRDF/CG — Consistency Groups

Note: For a detailed introduction to the Solutions Enabler, SYMCLI, and the Symmetrix array, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

SRDF provides a recovery solution for component or site failures between remotely mirrored devices. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster.

In an SRDF configuration, the individual Symmetrix devices are designated as either a *source* or a *target* to synchronize and coordinate SRDF activity. If the source (R1) device fails, the data on its corresponding target (R2) device can be accessed.

When the source (R1) device is replaced, the source (R1) device can be resynchronized. SRDF configurations have at least one source (R1) device mirrored to one target (R2) device.

Most operations described in this manual require an SRDF/Synchronous license unless otherwise specified.

Note: The terms SRDF and RDF are used throughout this manual and refer to the Symmetrix Remote Data Facility.

SRDF Site Configurations

SRDF site configurations provide for either a unidirectional or a bidirectional data transfer from one storage site to another. In a unidirectional SRDF configuration, all source (R1) devices reside in the local Symmetrix array and all target (R2) devices in the remote site Symmetrix array. Data flows from the source (R1) devices over an SRDF link to the target (R2) devices.

In a bidirectional configuration, both source (R1) and target (R2) devices reside in each Symmetrix array, as the master copy point and the mirror copy point, in the SRDF configuration. Data flows from the source (R1) devices to the target (R2) devices.

Figure 1-1 on page 1-4 illustrates the SRDF bidirectional configuration where both source (R1) and target (R2) devices reside within a Symmetrix array. Data flows from the source (R1) device in each respective Symmetrix array to the target (R2) device. If host A is the controlling point, the locally connected Symmetrix array has the R1 device. If host B at the remote site is the controlling point, its locally connected Symmetrix array has the R1 device.

A source (R1) device can only belong to a device/composite group of type RDF1, while a target (R2) device can only belong to these groups of type RDF2.

Note: In this chapter, each source (R1) device and its corresponding target (R2) device form an RDF pair.

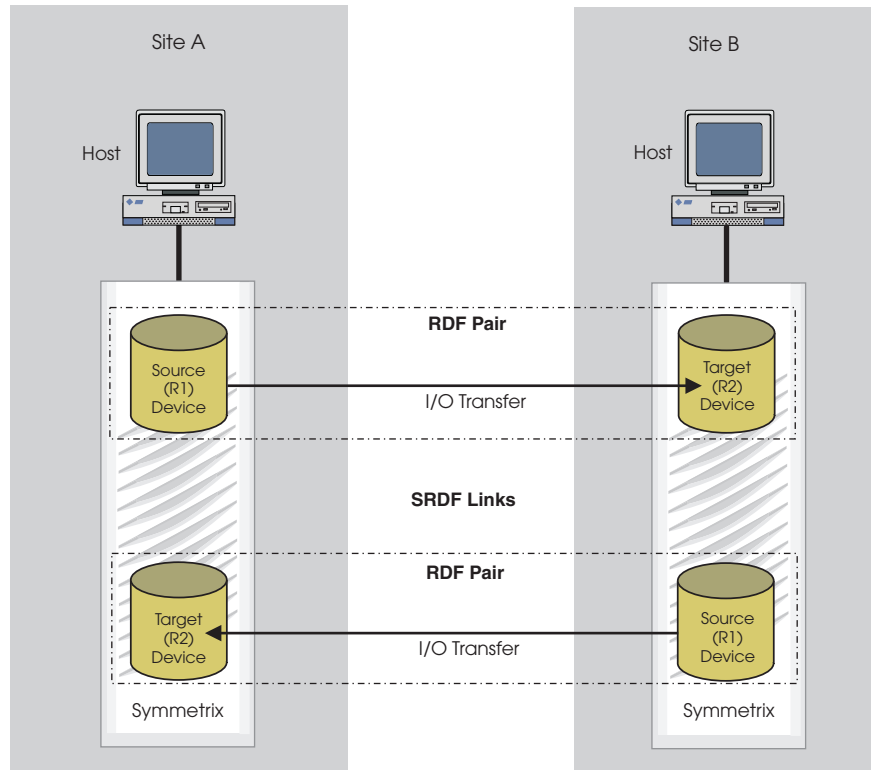


Figure 1-1 SRDF Bidirectional Configuration

SRDF Types of Implementation

The two types of SRDF point-to-point network implementations are the *Campus Solution* and the *Extended Distance Solution*. The SRDF Campus Solution shown in Figure 1-2 allows Symmetrix arrays to be located up to 60 km (37.5 miles) apart using fiber-optic links and even farther (thousands of miles) with extended-distance solutions such as FarPoint™.

Synchronous, asynchronous (SRDF / A), semi-synchronous, or adaptive copy operational modes are available for extended distance solutions. The Campus Solution supports both unidirectional and bidirectional SRDF connections.

Note: For more information on the available SRDF operational modes, refer to Chapter 2, *Setting SRDF Modes* on page 2-54.

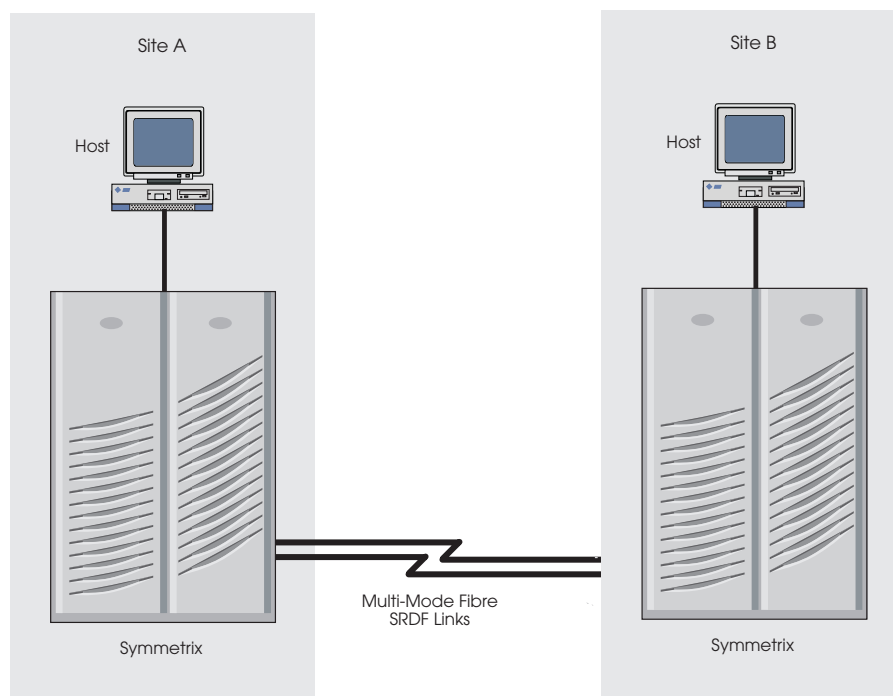


Figure 1-2 SRDF Campus Solution

Note: Using a T1/T3 or E1/E3 links, an SRDF Extended Distance Solution allows the Symmetrix arrays to be located over 37.5 miles (60 km) apart.

Switched RDF Environments

An SRDF topology can incorporate open network switching (*fabric*) in the SRDF links (refer to Figure 1-3). The switched RDF involves non-blocking switching devices that interconnect two or more nodes. Symmetrix arrays in a switched RDF topology can have each port pair running full duplex.

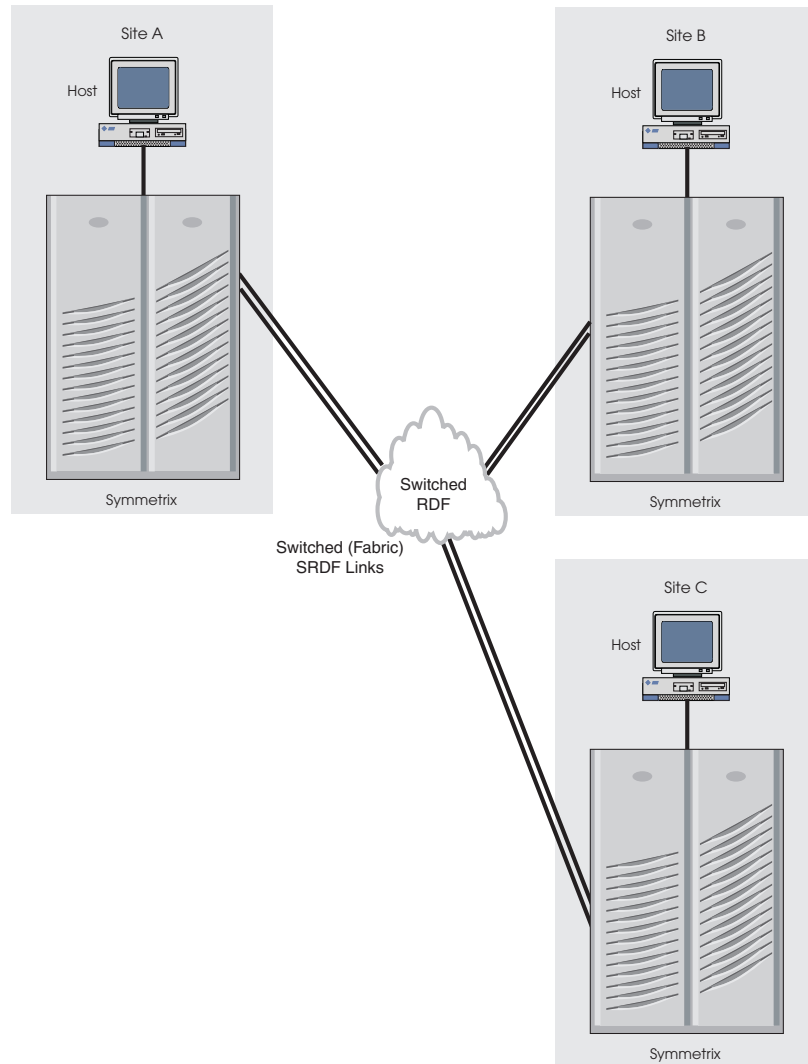


Figure 1-3 Switched (Fabric) RDF Topology

SRDF Device Types and Groups

This section describes the devices types and device groups concepts that are specific to SRDF configurations.

SRDF Devices

When configured for SRDF, the individual Symmetrix devices are designated as either a *source* (R1 device) or a *target* (R2 device) to synchronize and coordinate remote mirroring activities. If the source device fails, the data on its corresponding target device can be accessed by the local host. Once the source device is replaced, it can be resynchronized.

SRDF configurations have at least one source (R1) device mirrored to one target (R2) device. For concurrent RDF systems, there can be two R2 targets. A source (R1) device can only belong to an RDF1 device group, while a target (R2) device can only belong to an RDF2 device group.

Dynamic RDF Devices

Since Enginuity™ 5568, devices can be configured to be dynamic RDF-capable devices. Dynamic RDF functionality enables you to create, delete, and swap RDF pairs while the Symmetrix array is in operation. Using dynamic RDF technology, you can establish RDF device pairs from non-configured RDF devices, and then synchronize and manage them in the same way as configured SRDF pairs.

The dynamic RDF configuration state of the Symmetrix array must be enabled in SymmWin or via the Configuration Manager and the devices must be designated as dynamic RDF-capable devices.

For information about dynamic SRDF devices, refer to *Dynamic SRDF Pair Operations* on page 2-59.

RDF Device Groups

An RDF device group is a user-defined device group comprised of RDF devices belonging to a single Symmetrix array. At the time of creation, a device group must be defined as type REGULAR, RDF1, or RDF2. If the group type is defined as RDF1 or RDF2, the group is considered an RDF group. A device cannot belong to more than one device group. You can use device groups to identify and work with a subset of available Symmetrix devices, obtain configuration, status, and performance statistics on a collection of related devices, or issue control operations that apply to all devices in the specified device group.

Composite Groups

A *composite group* (CG) is a user-defined group of device members that can span multiple Symmetrix arrays and RDF groups. The CG type may be defined as REGULAR, RDF1, or RDF2 and may contain various device lists for standard, BCV, virtual (VDEV), and remote devices. Refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide* for information on composite groups and using the `symcg` command.

SRDF Consistency Groups

An *SRDF consistency group* (SRDF/CG) is a composite group comprised of RDF devices (RDF1 or RDF2), which has been enabled for remote database consistency. The RDF consistency groups operate in unison to preserve the integrity and dependent write consistency of a database distributed across multiple arrays. Depending on your mode of operation, consistency is maintained via PowerPath® or Multi Session Consistency (used for asynchronous operations), which respects the logical relationships between dependant I/Os. Consistency group functionality requires an SRDF/CG license and an additional SRDF/A (asynchronous) license to enable consistency groups running in asynchronous mode. RDF consistency group configurations are controlled via the host over SRDF links.

When a typical DBMS application updates a database, it first writes to the disk containing a log, and then it writes the data to the actual database datafiles. Finally, it writes again to the log volume to flag these write I/Os (log database) that are related.

Even in a remote disk copy environment, data consistency cannot be ensured if one of these I/Os was remotely mirrored, but its predecessor was not remotely mirrored. This could occur, for example, in a *rolling* disaster where there is a communication loss that affects only a portion of the disk controllers that are performing the remote copy function.

Consistency groups can prevent this from occurring by intercepting any I/O to a disk device that cannot communicate to its remote mirror. The consistency protocol is to then suspend the remote mirroring for all devices defined to the consistency group. In this way, consistency groups prevent dependent I/O from getting out of sync, thus ensuring the integrity and consistency of the data at the remote site.

Note: Refer to Chapter 3, *RDF Consistency Group Operations* on page 3-55 for information on enabling consistency for composite groups.

Command Summary

The SRDF component commands allow you to perform control operations on RDF devices. Using the `symrdf` command, you can perform operations including:

- ◆ Setting the RDF mode for one or more RDF pairs in a device group, composite group, or file.
- ◆ Returning information about the state of RDF mirroring
- ◆ Pinging one or more Symmetrix arrays locally or remotely via SRDF links
- ◆ Running singular SRDF control operations, which are the individual operations that comprise the composite SRDF control actions
- ◆ Performing dynamic group operations

Note: For information about performing SRDF control operations, refer to Chapter 2. For syntax and details about the SYMCLI commands, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Table 1-1 summarizes the actions of the SRDF commands.

Table 1-1 **SRDF Command Summary**

Command	Description
<code>symioctl</code>	Sends I/O control commands to a specified server application. • Places objects into hot backup mode. • Freezes or thaws I/O to a specified database application. • Issues a checkpoint to the RDBMS. • Archives the current log. • Begins a snapshot backup on SQL Server 2000 and higher. • Saves snapshot meta data and resumes writes on SQL Server 2000 and higher. • Restores previously saved snapshot meta-data on SQL Server 2000 and higher. • Terminates the snapshot operation without saving meta data, and resumes writes on SQL Server 2000 and higher. (continued)

Table 1-1 SRDF Command Summary (continued)

Command	Description
<code>symrdf</code>	Performs the following control operations on RDF devices: - Establishes (mirrors) an RDF pair by initiating a data copy from the source (R1) side to the target (R2) side. This operation can be a full or incremental establish. - Restores remote mirroring. Initiates a data copy from the target (R2) side to the source (R1) side. This operation can be a full or incremental restore. - Splits an RDF pair, which stops mirroring for the RDF pair(s) in a device group. - Fails over from the source (R1) side to the target (R2) side, switching data processing to the target (R2) side. - Fails back from the target (R2) side to the source (R1) side, switching data processing to the source (R1) side. - Updates the source (R1) side after a failover, while the target (R2) side may still be operational to its local host(s). - Swaps the source (R1) and target (R2) destinations between the target and the source. - Creates, deletes, or swaps dynamic SRDF device pairs. - Performs dynamic RDF group controls to add, modify, and remove dynamic groups. - Enables link domino locally or remotely when creating dynamic groups. - Enables auto link recovery locally or remotely when creating dynamic groups. - Enables/disables consistency for SRDF/A capable devices operating in asynchronous mode that are managed by a device group or file.
<code>symreplicate</code>	The SRDF/AR command that invokes a replicate session that generates automated recurrent, background copies of the standard data following a path across SRDF links and cascading BCVs. You can start a replicate session, stop it, and restart the replicate session.

This chapter identifies the Symmetrix command line interface (SYMCLI) actions and specific commands required to manage the Symmetrix Remote Data Facility (SRDF). It focuses on the various arguments, options, and the application of certain parameters for the SRDF monitor and control actions. Using examples of SRDF commands, it describes how to manage the behavior and states of the various SRDF components in a typical configuration.

The terms SRDF and RDF are used throughout this chapter and refer to the Symmetrix Remote Data Facility.

◆ Finding SRDF Devices	2-2
◆ SRDF Operations	2-5
◆ Composite SRDF Control Operations	2-6
◆ Singular SRDF Control Operations	2-29
◆ SRDF States	2-47
◆ RDF Operations and Applicable States	2-50
◆ Setting SRDF Modes	2-54
◆ Dynamic SRDF Pair Operations	2-59
◆ Dynamic R1/R2 Swap	2-67
◆ Dynamic Failover Establish	2-70
◆ symrdf -g Command Line Options	2-72
◆ symrdf -cg Command Line Options	2-81
◆ symrdf -file Command Line Options	2-84

Finding SRDF Devices

Configuration and status information can be viewed for each device on every Symmetrix array containing SRDF devices.

Using SYMCLI, you can find all SRDF devices on a Symmetrix array and view their physical (host) and Symmetrix device names. In addition, you can display details about the SRDF devices, the number of invalid tracks for both the SRDF source device and the target device, and the various SRDF device states.

You can find all Symmetrix arrays that are reachable via the SRDF links. For example, to view how Symmetrix arrays are attached to your host, enter:

```
symcfg list
```

Device Groups

The `symdg list` command lists the device groups by name, and also by group type. To view all device groups that have been created in your host database file, enter:

```
symdg list
```

If the type of group is RDF1, or RDF2, the group is an SRDF device group. In SRDF environments, these are referred to as an RDF group or an RA group.

Composite Groups

The `symcg list` command lists the composite groups by name, and also by group type. To view all composite groups that have been created in your host database file, enter:

```
symcg list
```

If the type of group is RDF1, or RDF2, the group is an SRDF composite group. A composite group may contain one or more RDF group or RA group.

Listing SRDF Devices

The `symrdf list` command lists the SRDF devices that are visible to your host, or SRDF devices that are configured on a given Symmetrix array.

For example, to list the SRDF devices that are visible to your host, enter:

```
symrdf list pd
```

The results provide details about the SRDF devices, source (R1) and target (R2), their device groups (if any), including:

- ◆ Symmetrix device name
- ◆ Remote Symmetrix device name
- ◆ RDF type and RA number
- ◆ Status of the SA, RA, and SRDF links
- ◆ SRDF mode
- ◆ Domino mode
- ◆ Adaptive copy mode
- ◆ Number of invalid tracks on R1 and R2
- ◆ SRDF link state
- ◆ RDF states on the device, remote device, and RDF pair

For example, to obtain detailed information on all SRDF devices in Symmetrix array 123, enter:

```
symrdf -v -sid 123 list dev
```

By default, the `symrdf list` command provides a listing of all SRDF devices available, including SRDF BCV devices.

```
symrdf list
```

To display only the SRDF standard devices in Symmetrix array 123, enter:

```
symrdf -sid 123 -nobcv list
```

You can display the R1 devices separately, or the R2 devices separately, by using the `-R1` or `-R2` option. For example, if you want to display only the SRDF R1 devices in Symmetrix array 123, enter:

```
symrdf -sid 123 -R1 list
```

To display only the SRDF devices capable of operating in asynchronous mode in Symmetrix array 123, enter:

```
symrdf -sid 123 -rdfa list
```

Note: Beginning with Enginuity 5671, all devices are SRDF/A-capable and the command will display all devices.

Query SRDF Devices

After executing any SRDF control operation, use the `symrdf query` command to verify the results and impact on selected devices and device groups. The `symrdf query` argument can be used with the `-g DgName`, `-cg CgName` and `-file FileName` options. Refer to the *EMC Solutions Enabler Symmetrix Command Reference* manual for specific information of `symrdf` syntax and options available.

To view the SRDF details about all devices in device group `prod`, enter:

```
symrdf -g prod query
```

The query results provide details about each RDF pair in the device group, including:

- ◆ Logical device name
- ◆ Physical device name
- ◆ Number of invalid tracks on R1 and R2
- ◆ SRDF link state
- ◆ RDF modes
- ◆ RDF local and remote SRDF states
- ◆ RDF pair state

Ping SRDF Devices

You can use the `symrdf -rdf ping` option to determine if a Symmetrix array that is connected via SRDF links is up and running. The Symmetrix array(s) are pinged via the SRDF links. Based on return codes, you can determine whether some or all of the Symmetrix arrays were successfully pinged.

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

For example, to ping Symmetrix array 123 via the SRDF links, enter:

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

Verify SRDF States

You can verify that the RDF pairs are in the Synchronized or Restored states.

For example, to verify that the RDF pair `DEV007`, in device group `prod`, is in the Synchronized state, enter:

```
symrdf -g prod verify DEV007 -synchronized
```

SRDF Operations

SRDF control operations, which are performed with the `symrdf` command, support the high-level operations of the SRDF environment, such as failover (disaster recovery), backup or copy (decision support), and concurrent operations.

Types of SRDF Control Operations

The SYMCLI `symrdf` command performs these operations with low-level control operations, which are defined as two types: *composite* and *singular* operations. Several singular operations may make up a composite operation.

Note: Most SRDF operations can be performed with just composite SRDF control operations. It is recommended you use the singular control operations sparingly.

The composite SRDF control operations are described next. The singular SRDF control operations are described in *Singular SRDF Control Operations* on page 2-29.

Migrating Data from R1 to a Larger R2 Device

Enginuity 5669 provides a temporary method for migrating data from an R1 device to a larger R2 device. Data migration to a larger R2 device is only allowed on Symmetrix DMX arrays running Enginuity 5669 or higher.

The following SRDF operations are blocked when the R2 device is larger than the R1 device:

- ◆ Swap operations are not allowed if the R2 device is larger than the R1 device.
- ◆ Data migrated to a larger R2 device cannot be restored back to the R1 device.
- ◆ Concatenated meta devices are not supported; striped meta devices are supported.

Note: Depending on the type of file system and attached host, certain host-dependent operations may be required to access the migrated data.

Composite SRDF Control Operations

To manage SRDF devices, you will need to invoke composite control operations of the `symrdf` command on a device or composite group of remotely mirrored devices, such as a device/composite group of type RDF1 or RDF2.

Most operations described in this chapter require an SRDF or SRDF/Synchronous license unless otherwise specified.

Table 2-1 outlines the composite SRDF control operations, the corresponding `symrdf` action argument, and the results of implementing the `symrdf` command.

Table 2-1 Composite SRDF Control Operations

Control Operation	<code>symrdf</code> Action Arguments	Results
Establishing an RDF pair	<code>establish -full</code>	Establishes remote mirroring and initiates a full data copy from the source (R1) device to the target (R2) device.
Incrementally Establishing an RDF pair	<code>establish</code>	Establishes remote mirroring and initiates an incremental data copy from the source (R1) device to the target (R2) device.
Restoring from a target (R2) device	<code>restore -full</code>	Resumes remote mirroring and initiates a full data copy from the target (R2) device to the source (R1) device.
Incrementally Restoring from a target (R2) device	<code>restore</code>	Resumes remote mirroring and initiates an incremental data copy from the target (R2) device to the source (R1) device.
Splitting an RDF pair	<code>split</code>	Stops remote mirroring between the source (R1) device and the target (R2) device. The target device is made available for local host operations.
Failover	<code>failover</code>	Switches data processing from the source (R1) side to the target (R2) side.
Failback	<code>failback</code>	Switches data processing from the target side (R2) back to the source (R1) side.
Update R1 mirror	<code>update</code>	Updates the source (R1) side with the changes from the target (R2) side while the target (R2) side is still operational to its local host(s).

Table 2-1 Composite SRDF Control Operations (continued)

Control Operation	symrdf Action Arguments	Results
Swap R1 designations with R2 types	swap	Swaps the source (R1) designations with the target (R2) designations.
Create SRDF Pairs	createpair	Creates the dynamic SRDF pairs specified in the device file.
Delete SRDF Pairs	deletepair	Deletes the SRDF pairs specified in the device file.

The SRDF control operations listed in Table 2-1 invoke several singular SRDF control operations, which are listed in Table 2-3 on page 2-31.

The composite SRDF control operations outlined in Table 2-1 are described in the following pages of this section.

Device External Locks

SYMAPI/SYMCLI uses *device external locks* in the Symmetrix array to lock BCV pairs during TimeFinder control operations and to lock RDF device pairs during SRDF control operations.

To list a range of Symmetrix devices (0000 to 000A) that have a device external lock, enter:

```
symdev list -sid 870 -RANGE 0000:000A -lock
```

On your host, if you discover a lock that has been on for well over 2 hours and are sure no one is using the locked device resources, you can choose to release the lock. To release the device lock on a range of Symmetrix devices in Symmetrix 870, enter:

```
symdev release -sid 870 -RANGE 0000:000A
```



CAUTION

Use the release lock action only if you believe that the Symmetrix device lock was forgotten and there are no other operations in progress to the specified Symmetrix devices (local or remote). Locks are typically short duration (one second to an hour or so). But, be ready to recognize when a device lock being held by a certain application (such as an RDF action) might be allocated as a long-duration lock. Also be aware that device external locks are held during the entire replication session for devices participating in an SRDF/AR (Automated Replication) session.

Disallow Synchronization Actions

For some sites, it may be desirable to block users on a host from performing either an establish or restore operation on any of the Symmetrix devices. The sync direction parameter (`SYMAPI_SYNC_DIRECTION`) in the options file allows you to confine SRDF and TimeFinder operations to only establish or restore actions.

You can block a user on a host from executing a restore or an establish action using the following form:

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

`ESTABLISH` confines the possible operations to just establish actions.

`RESTORE` confines the possible operations to just restore actions, which includes (allows) restore, failback, R1 update actions.

`BOTH` is the default, which does not restrict any SRDF or TimeFinder actions.

Locking at the RA Group Level Instead of Symmetrix

In the options file, the parallel RA groups parameter (`SYMAPI_PARALLEL_RA_GROUPS`) allows you to lock RA groups (RDF groups) during RDF control actions, instead of applying Symmetrix-wide locks. This enables concurrent RDF control actions to be done at the same time (parallel) across different RA groups. To enable this parallel RA group feature, enter:

```
SYMAPI_PARALLEL_RA_GROUPS = ENABLE
```

`DISABLE` is the default.

You may have up to 16 RA groups that can execute RDF control actions in parallel. With Enginuity 5669 and above you may have up to 64 RA groups that can execute RDF control actions in parallel.

Note: Since Enginuity 5669, locking at the Symmetrix array and RA group level for non-dynamic RDF operations has been removed. Locking for non-dynamic RDF operations is now provided at the device level.

SRDF Operations and Copy Sessions

Certain SRDF operations are not allowed within Symmetrix 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.

Refer to Appendix A, *TimeFinder/Snap and Clone State Reference*, for a description of the TimeFinder/Snap and TimeFinder Clone pair states and which SRDF operations are available within each state.

Full Establish

You need to perform a *full establish* on RDF pairs only when you are initially setting up RDF pairs, or when your R2 member of an RDF pair is either fully invalid, or has been replaced. All the RDF pairs must be in the split state before you establish the pairs.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to establish. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock, or to release device locks, refer to *Device External Locks* on page 2-7.

When the establish control operation has successfully completed and the device pair has fully synchronized, the RDF pairs will contain identical data. You can use `verify` to confirm that the RDF pair(s) are in the Synchronized state and remote mirroring is resumed.

The full establish control operation can be performed by device group, composite group, or device file:

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

Note: For more detail about defining a device file, refer to *Device File* on page 2-84.

For example, to initiate an establish for all the RDF pairs in the device group `prod`, enter:

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

To initiate an establish for one RDF pair with logical device `DEV001` in the device group `prod`, enter:

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

To initiate an establish for a list of RDF pairs in the device group `prod`, enter:

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

To initiate a full establish, all RDF pairs in the group must already be in one of the following states:

- ◆ Split
- ◆ Suspended and Write Disabled or Not Ready at the source
- ◆ Invalid, R1 and R2 are Not Ready and the link is Ready
- ◆ R1 Updated or Failed Over and the R1 is not visible to any host

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 Symmetrix Base Management CLI Product Guide*.

Figure 2-1 illustrates the establishing of an RDF pair. The RDF pair consists of the source (R1) device that is mirrored to the target (R2) device.

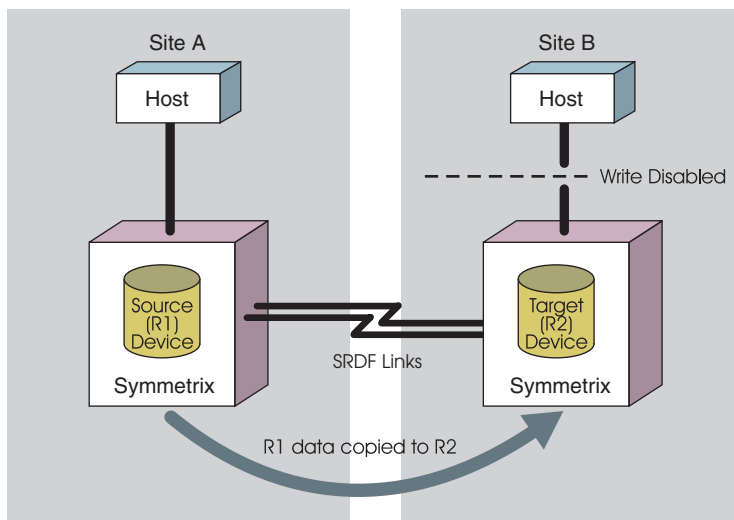


Figure 2-1 Establishing an RDF Pair

When a full establish is initiated for each specified RDF pair in a device group:

- ◆ The target (R2) device is Write Disabled to its local host(s).
- ◆ Traffic is suspend 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.

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

Note: This operation will be rejected if the source has invalid local (R1) tracks.

Incremental Establish

Incrementally establishing an RDF pair (Figure 2-2 on page 2-12) accomplishes the same thing as the establish process, with a major time-saving exception: the source (R1) device copies to the target (R2) device only the new data that was updated on the source (R1) device while the RDF pair was split. Additionally, any data that was modified on the target (R2) device will be refreshed from the corresponding tracks on the source (R1) side.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to establish. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

When the establish control operation has successfully completed and the SRDF device pair has synchronized, the RDF pairs will contain identical data.

The incremental establish control operation can be performed by device group, composite group, or device file:

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

Note: For more detail about defining a device file, refer to *Device File* on page 2-84.

For example, to initiate an incremental establish on all RDF pairs in the `prod` device group, enter:

```
symrdf -g prod establish
```

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

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

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

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

To invoke this operation, the RDF pair(s) must already be in one of the following RDF states:

- ◆ Split
- ◆ Suspended
- ◆ Invalid, R1 and R2 are Not Ready and the link is Ready

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 Symmetrix Base Management CLI Product Guide*.

Figure 2-2 illustrates the incremental establishing of an RDF pair. The RDF pair consists of the source (R1) device that is mirrored to the target (R2) device.

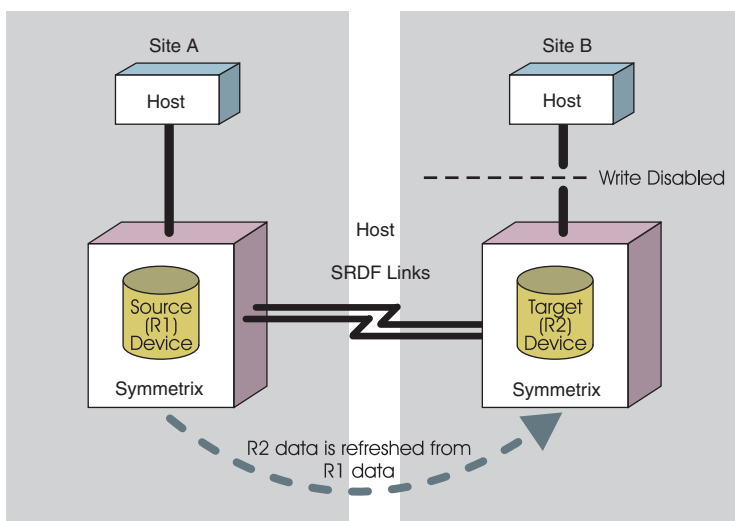


Figure 2-2 Incremental Establish of an RDF Pair

When an incremental establish is initiated for each specified RDF pair in a device group:

- ◆ The target (R2) device is Write Disabled to its local host(s).
- ◆ Traffic is suspend on the SRDF links.
- ◆ The invalid tracks on the target (R2) device are refreshed from the changed tracks of the source (R1) device.
- ◆ The track tables are merged between the source (R1) device and the target (R2) device.
- ◆ Traffic is resumed on the SRDF links.

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

Note: An incremental establish will be rejected if the source has invalid local (R1) tracks.

Split

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

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to split. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

The split control operation can be performed by device group, composite group, or device file:

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

Note: For more detail about defining a device file, refer to *Device File* on page 2-84.

For example, to perform a split on all the RDF pairs in the `prod` device group, enter:

```
symrdf -g prod split
```

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

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

To initiate a split to a list of RDF pairs in the device group `prod`, enter:

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

To invoke a split, the RDF pair(s) must already be in one of the following states:

- ◆ Synchronized
- ◆ Suspended
- ◆ R1 Updated
- ◆ SyncInProgress and the `-force` option is specified

Figure 2-3 illustrates the splitting of an RDF pair. The RDF pair consists of the source (R1) device, which is remotely mirrored to the target (R2) device.

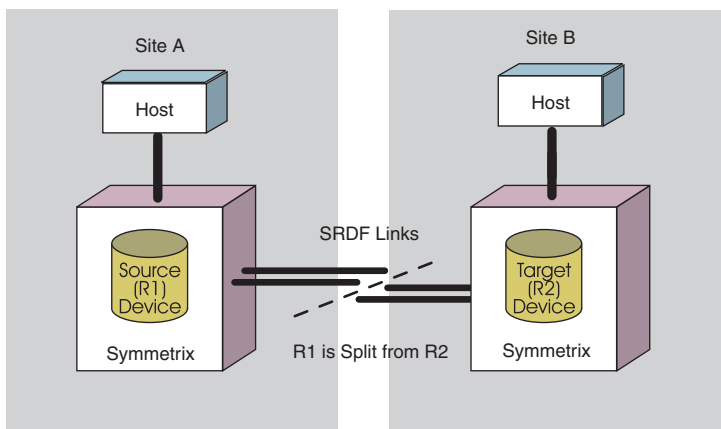


Figure 2-3 Splitting an RDF Pair

When a split is performed for each specified RDF pair in a device group:

- ◆ Traffic is suspended on the SRDF links.
- ◆ The target (R2) device is read/write enabled to its local host(s).

After the target (R2) device is split from the source (R1) device, the RDF pair is in the Split state.

Note: This operation will be rejected if any of the following occur:

- The source has invalid local (R1) tracks.
 - The target has invalid local (R2) tracks.
 - The device pairs are in the device domino or adaptive copy mode and the `-force` option is **not** specified.
 - The source has invalid remote (R2) tracks and the `-force` option is **not** specified.
 - Consistency is enabled and the `-force` option is **not** specified.
-

Splits Impacting Databases

If the SRDF split will impact the access integrity of a database, additional actions such as freezing the database to user access may be necessary. The freeze action can be used in conjunction with the TimeFinder or SRDF split operation. The freeze suspends the database updates being written to disk.

Using the `symioctl` command, you can invoke I/O control operations to freeze access to a specified relational database or database object(s).

Note: First, you must set `SYMCLI_RDB_CONNECT` to your username and password for access to the specified database.

Freeze

To freeze all I/O access to a specified relational database, you use the following command:

```
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 system.

If you have set the connection environment variables, you just need to enter:

```
symioctl freeze Object Object
```

For example, to freeze databases HR and Payroll, enter:

```
symioctl freeze HR Payroll
```

Thaw

Once the freeze action is completed, the split may proceed. When the split operation completes, a `symioctl thaw` command must be sent to resume full I/O access to the database instance. For example:

```
symioctl thaw
```

Hot Backup Control

For Oracle only, you can perform hot backup control on a list of tablespace objects, which must be performed before and after a freeze/thaw command. The steps required to split a group of RDF pairs follows:

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

Consistency Groups

For consistency group split operations, refer to *RDF Consistency Group Operations* on page 3-55.

Full Restore

The *full restore* operation differs from the establish operations in that the entire contents of the target (R2) device is copied to the source (R1) device. After the restore control operation has successfully completed, the RDF pairs will synchronize. You can use `verify` to confirm that the RDF pair(s) are in the Synchronized state.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to restore. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

The full restore operation can be performed by device group, composite group, or device file:

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

Note: For more detail about defining a device file, refer to *Device File* on page 2-84.

For example, to initiate a full restore on all RDF pairs in the `prod` device group, enter:

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

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

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

To initiate a full restore a list of RDF pairs in the device group `prod`, enter:

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

To invoke this operation, the RDF pair(s) must already be in one of the following RDF states:

- ◆ Split
- ◆ Suspended and Write Disabled at the source
- ◆ Suspended and Not Ready at the source
- ◆ Invalid, R1 and R2 are Not Ready and the link is Ready

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 Symmetrix Base Management CLI Product Guide*.

Figure 2-4 illustrates the restoring of an RDF pair. The RDF pair consists of the source (R1) device, mirrored to the target (R2) device.

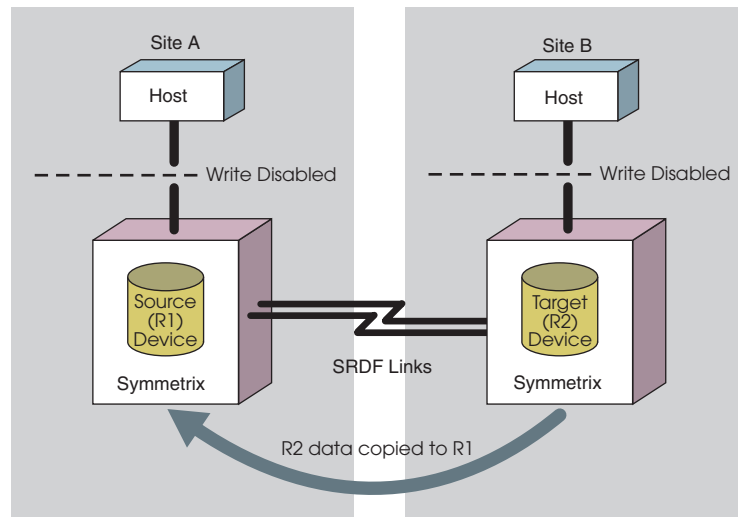


Figure 2-4 Restoring an SRDF Device

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

- ◆ The source (R1) device is Write Disabled to its local host(s).
- ◆ The target (R2) device is Write Disabled to its local host(s).
- ◆ Traffic is suspend 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 host(s).

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

Note: This operation will be rejected if the target has invalid local (R2) tracks.

Incremental Restore

The *incremental restore* process accomplishes the same thing as the restore process with a major time-saving exception: the target (R2) device copies to the source (R1) device only the new data that was updated on the target (R2) device while the RDF pair was split. Any changed tracks on the source (R1) device are refreshed from the corresponding tracks on the target (R2) device. After the restore control operation has successfully completed, the RDF pairs will synchronize. You can use `verify` to confirm that the RDF pair(s) are in the Synchronized state.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to restore. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

This process is useful if the results from running a new application on the target (R2) device were desirable, and the user wants to move the changed data and the new application to the source (R1) device.

The incremental restore operation can be performed by device group, composite group, or device file:

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

Note: For more detail about defining a device file, refer to *Device File* on page 2-84.

For example, to initiate an incremental restore on all RDF pairs in the `prod` device group, enter:

```
symrdf -g prod restore
```

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

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

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

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

To invoke this operation, the RDF pair must already be in one of the following states:

- ◆ Split
- ◆ Suspended and Write Disabled at the source
- ◆ Suspended and Not Ready at the source
- ◆ Suspended and the force (`-force`) option is specified
- ◆ Invalid, R1 and R2 are Not Ready and the link is Ready

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 Symmetrix Base Management CLI Product Guide*.

Figure 2-5 on page 2-20 illustrates the incremental restore of an RDF pair. The RDF pair consists of the source (R1) device, which is mirrored to the target (R2) device.

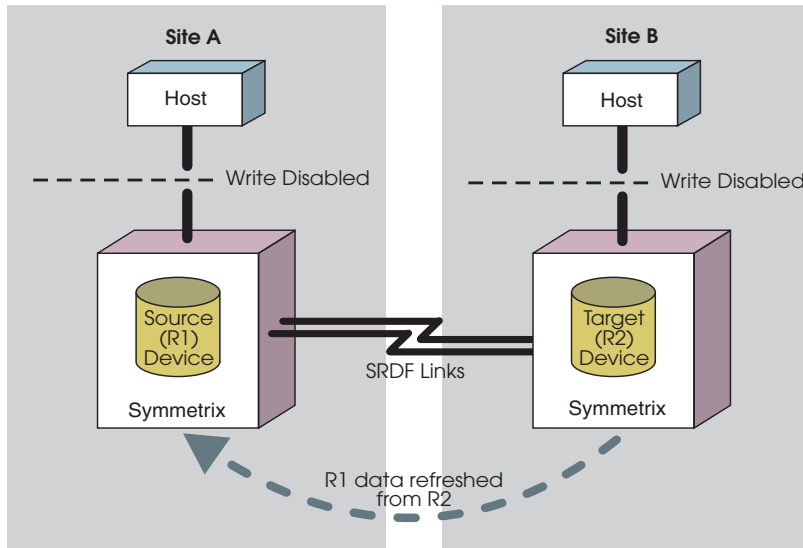


Figure 2-5 Incrementally Restoring an SRDF Device

When an incremental restore is initiated for each specified RDF pair in a device group:

- ◆ The source (R1) device is Write Disabled to its local host(s).
- ◆ The target (R2) device is Write Disabled to its local host(s).
- ◆ Suspend 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 host(s).

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

Note: This operation will be rejected if the target has invalid local (R2) tracks.

Failover

In a period of scheduled downtime for maintenance, or after a serious system problem which has rendered either the host or Symmetrix unit containing the source (R1) devices unreachable, no read/write operations can occur on the source (R1) device. In this situation, the *failover* operation should be initiated to make the target (R2) devices read/write enabled to their local host(s). Figure 2-6 on page 2-22 describes the failover procedure.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to fail over. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

The failover control operation can be performed by device group, composite group, or device file:

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

For example, to perform a failover on all the RDF pairs in the `prod` device group, enter:

```
symrdf -g prod failover
```

To perform a failover on one RDF pair with device `DEV001` in the `prod` device group, enter:

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

To perform a failover on a list of RDF pairs in the device group `prod`, enter:

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

To invoke a failover, the RDF pair(s) must already be in one of the following states:

- ◆ Synchronized
- ◆ Suspended
- ◆ R1 Updated
- ◆ Partitioned (when you are invoking this operation from the target side)

Note: This operation will be rejected if any of the device pairs are in the following states without specifying the `-force` option:

- Split
- SyncInProgress
- R1 UpdInProgress
- Invalid

Figure 2-6 illustrates the failover of an RDF pair. The RDF pair consists of the source (R1) device, which is mirrored to the target (R2) device.

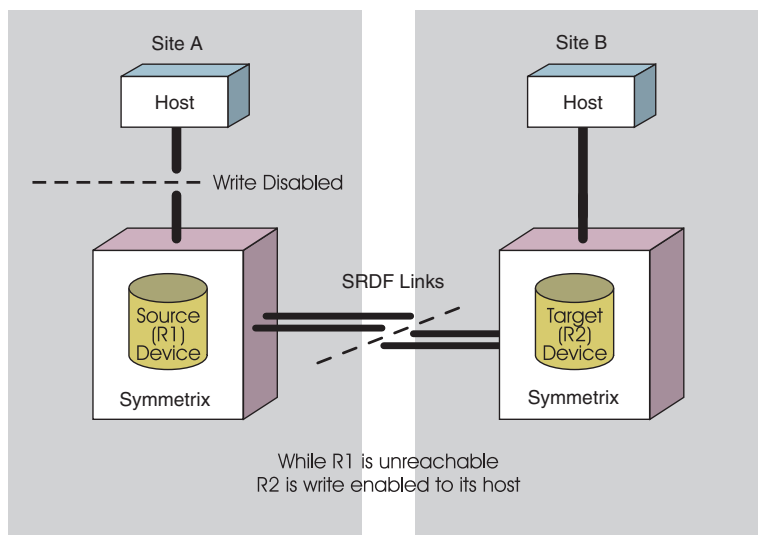


Figure 2-6 Failover of an SRDF Device

When a failover is performed for each specified RDF 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 host(s).

- ◆ The target (R2) device is Read/Write Enabled to its local host(s).

Note: This operation will be rejected if any of the following occur:

- If the source has invalid remote (R2) tracks without specifying the `-symforce` option.
 - If the target has invalid local (R2) tracks without specifying the `-symforce` option.
 - If consistency is enabled and the `-force` option is **not** specified.
-

Failback

A *failback*, or source (R1) device takeover, is performed when you are ready to resume normal SRDF operations by initiating read/write operations on the source (R1) devices, and stopping read/write operations on the target (R2) devices. The target (R2) devices become read-only to their local host(s) while the source (R1) devices are read/write enabled to their local host(s).

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to *failback*. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

The failback control operation can be performed by device group, composite group, or device file:

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

For example, to initiate a failback on all the RDF pairs in the `prod` device group, enter:

```
symrdf -g prod failback
```

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

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

To initiate a failback on a list of RDF pairs in the device group `prod`, enter:

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

To invoke a failback, the RDF pair(s) must already be in one of the following states:

- ◆ Failed Over
- ◆ Suspended and Write Disabled at the source
- ◆ Suspended and Not Ready at the source
- ◆ R1 Updated
- ◆ R1 UpdInProg

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 Symmetrix Base Management CLI Product Guide*.

Note: This operation will be rejected if any of the device pairs are in the Partitioned state unless you invoke this operation from the source side and specify the `-force` option.

Figure 2-7 illustrates the failback of an RDF pair. The RDF pair consists of the source (R1) device which is mirrored to the target (R2) device.

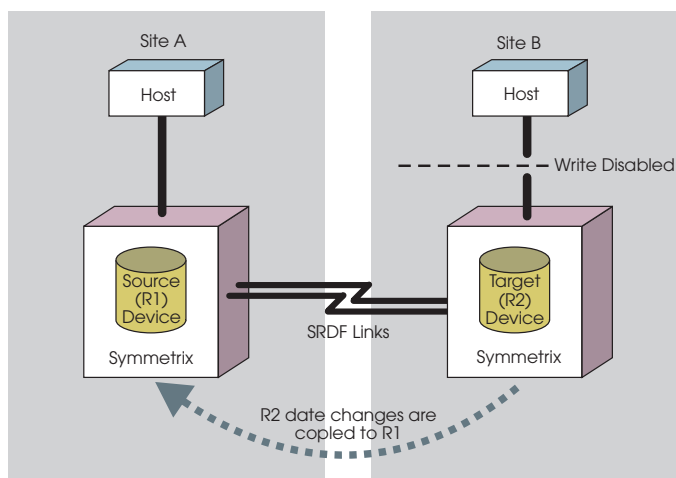


Figure 2-7 Failback of an SRDF Device

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

- ◆ The target (R2) device is Write Disabled to its local host(s).
- ◆ 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 host(s).

Note: This operation will be rejected if any of the following occur:

- If the source has invalid local (R1) tracks and the state is Partitioned.
 - If the source has invalid remote (R2) tracks and if the target side is not reachable. (If the target is reachable, use the `-force` option to mark the changed tracks on the source side to refresh from the target.)
 - If the target side is reachable and the target has invalid local (R2) tracks.
-

Update R1 Mirror

While the target (R2) device is still operational (Write Enabled to its local host(s)), an incremental data copy from the target (R2) device to the source (R1) device can be initiated in order to *update the R1 mirror* with changed tracks from the target (R2) device.

Note: When the `symrdf` command is initiated, device external locks are set on all RDF devices you are about to update. Device external locks are then automatically released when the control operation completes. For information on how to list a range of devices that have a device external lock or to release device locks, refer to *Device External Locks* on page 2-7.

The update R1 mirror control operations can be performed by device group, composite group, or device file:

```
symrdf -g DgName update
symrdf -cg CgName update
symrdf -f[file] FileName update
```

For example, to initiate an update of all the source (R1) devices in the RDF pairs, for device group `prod`, enter:

```
symrdf -g prod update
```

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

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

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

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

To invoke this operation, the RDF pair must already be in one of the following states:

- ◆ R1 Updated
- ◆ Failed Over
- ◆ Suspended and Write Disabled at the source
- ◆ Suspended and Not Ready at the source

Figure 2-8 illustrates the update of an RDF pair. The RDF pair consists of the source (R1) device which is mirrored to the target (R2) device.

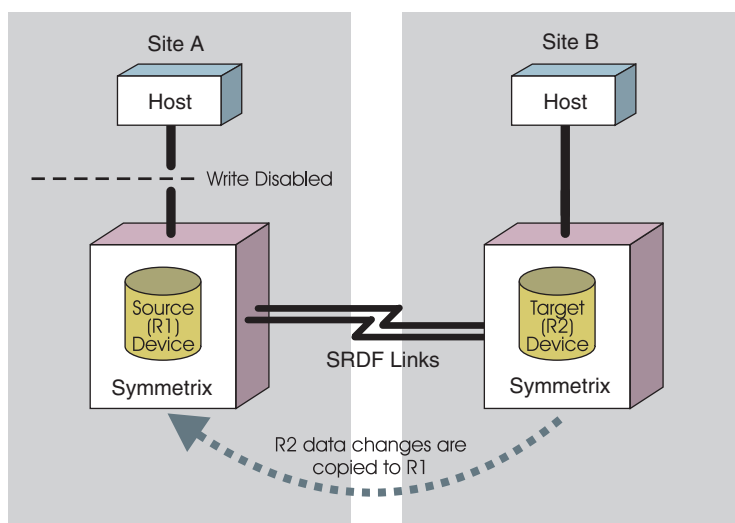


Figure 2-8 Update of SRDF Device Track Tables

An update is initiated for each specified RDF 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 device(s) 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.

When the update has completed successfully, the RDF pairs will be in the R1 Updated state.



CAUTION

When you perform an update while the RDF pair is Suspended and Not Ready at the source, the RDF pair enters an Invalid state as the update completes. To resolve this condition, you could then `rw_enable r1`, then the RDF pairs would become Synchronized.

Note: This operation will be rejected if the source has any invalid remote (R2) tracks and the `-force` option is **not** specified.

Continuous R1 Updates

You can perform continuous updates with one command (`update -until #`) for situations when you have or want I/O to continue via the remote host and periodically update an inactive R1 device over an extended period of time.

The `until (-until)` option when used with the `update` argument 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.

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

1. Update R1 mirror.
2. Changed tracks build up on R2.
3. Check invalid track count.

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

For example, to update the R1 mirror when track changes are in excess of 1000 on the R2, enter:

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

In this example, the R1 mirror will be continuously updated until the number of tracks to be copied is below 1000.

Create SRDF Pairs

The create SRDF pairs command creates SRDF pairs from devices listed in a device pairs file.

For example, to create SRDF pairs from a device pairs file called `devicefile`, enter:

```
symrdf createpair -sid 123 -file devicefile \
-type rdf1 -rdfg 10 -establish
```

For more information and specific options to the `symrdf createpair` command, refer to *Dynamic SRDF Pair Operations* on page 2-59.

Delete SRDF Pairs

The delete SRDF pairs command cancels SRDF pairs in the device file specified.

For example, to delete the SRDF pairs in a RDF group 10, enter:

```
symrdf deletepair -sid 123 -file devicefile -rdfg 10
```

For more detailed information on the `symrdf deletepair` command for device files, refer to *Delete Dynamic SRDF Pairs* on page 2-64.

The `deletepair` command can also be executed on device groups (`-g`) instead of specifying the device text file. Refer to *Control of Dynamic Pairs by Device Group* on page 2-66 for specific instructions.

Singular SRDF Control Operations

The singular SRDF control operations are invoked using `symrdf`. As shown in Table 2-2, the singular SRDF control operations make up the composite SRDF control operations. It is recommended that you use the composite SRDF control operations listed in Table 2-1 on page 2-6 before attempting to use the singular SRDF control operations.

Table 2-2 Decomposition of Composite Operations Into Singular Operations

Composite Operation	Individual Singular Operations	When Used
Full Establish	- Write Disable R2 devices on RA - Suspend RDF link traffic - Mark target device invalid - Merge track tables - Resume RDF link traffic	- Initial synchronization of RDF mirrors - Replacement of failed drive on the R2 side
Incremental Establish	- Write Disable R2 devices on RA - Suspend RDF link traffic - Refresh tracks on target - Merge track tables - Resume RDF link traffic	Resynchronization of RDF mirrors after they have been split and target data can be discarded
Split	- Suspend RDF link traffic - Read/Write Enable R2 to its local host	When both sides need to be independently accessible (e.g., for testing)
Full Restore	- Write Disable R1 to host - Write Disable R2 devices on RA - Suspend RDF link traffic - Mark all source tracks invalid - Merge track tables - Resume RDF link traffic - Read/Write Enable R1 to host	- Initial (reverse) synchronization of RDF mirrors - Replacement of failed drive on R1 side
Incremental Restore	- Write Disable R1 to host - Write Disable R2 devices on RA - Suspend RDF link traffic - Refresh source invalid tracks - Merge track tables - Resume RDF link traffic - Read/Write Enable R1 to host	Re-synchronize RDF mirrors after they have been split and the source can be discarded

Table 2-2 Decomposition of Composite Operations Into Singular Operations

Composite Operation	Individual Singular Operations	When Used
Failover	- Write Disable R1 to hosts - Suspend RDF link traffic - Read/Write Enable R2 to hosts	In the event of a failure of the source site
Failback	- Write Disable R2 on RA - Refresh source invalid tracks (requires use of <code>-force</code> option) - Merge track tables - Resume RDF link traffic - Write Enable R1 to hosts	To return to the source site from the target site after the cause of failure has been remedied
Update	- Suspend RDF link traffic - Refresh source invalid tracks (requires use of <code>-force</code> option) - Merge track tables - Resume RDF link traffic	To get the R1 site close to synchronized with the R2 side before a failback, while the R2 side is still online to the host

Invalid Tracks

Synchronizing SRDF devices is based on tracking and managing the changed tracks on a device with singular commands. The concept of invalid tracks in SRDF systems indicates what data is not synchronized between the two devices that form an SRDF pair. On both the source and target sides of an SRDF setup, the Symmetrix array keeps an account of the tracks that are "owed" to the other side. The owed tracks are known as remote invalids.

For example, consider the case of an R1 device whose logical connection to its R2 has been suspended. If both devices are made write-accessible, hosts on both sides of the RDF link can write to their respective devices, creating R2 invalids on the R1 side and R1 invalids 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 link between the R1 and R2, the invalid tracks have to be resolved.

The resolution of invalid tracks depends on which operation you perform. For instance, you can have remote invalids on both sides prior to an `establish` or a `restore` operation. If so, performing an `establish` operation indicates to SRDF that you want to copy modified R1 tracks to the R2 side. In the process, any tracks that were modified on the R2 side are overwritten with data from corresponding tracks on the R1 side.

Performing a `restore` operation indicates the opposite—that you want to copy modified R2 tracks to the R1 side. In the process, any tracks that were modified on the R1 side are overwritten with data from corresponding tracks on the R2 side.

Note: The singular SRDF control operations listed in Table 2-3 should be used sparingly, and only when all other composite control options have been exhausted.

Table 2-3 Singular SRDF Control Operations

Control Operation	<code>symrdf</code> Action Arguments	Results
Suspend SRDF links	<code>suspend</code>	Suspends I/O traffic on the SRDF links for the remotely mirrored RDF pair(s) in the group.
Resume SRDF links	<code>resume</code>	Resumes I/O traffic on the SRDF links for the remotely mirrored RDF pair(s) in the group.
Write Enable source device	<code>rw_enable r1</code>	Write Enables the source (R1) device to its local host.
Write Enable target device	<code>rw_enable r2</code>	Write Enables the target (R2) device to its local host.
Write Disable source device	<code>write_disable r1</code>	Write Disables the source (R1) device to its local host.
Write Disable target device	<code>write_disable r2</code>	Write Disables the target (R2) device to its local host.
Read/Write Disable target device	<code>rw_disable r2</code>	Read/Write Disables the target (R2) device to its local host.
Refresh R1 mirror	<code>refresh r1</code>	Marks any changed tracks on the source (R1) side to be refreshed from the R2 side.
Refresh R2 mirror	<code>refresh r2</code>	Marks any changed tracks on the target (R2) side to be refreshed from the R1 side.
Invalidate R1 mirror	<code>invalidate r1</code>	Invalidates all tracks on the source (R1) side so that they can be copied over from the target (R2) side.
Invalidate R2 mirror	<code>invalidate r2</code>	Invalidates all tracks on the target (R2) side so that they can be copied over from the source (R1) side.
Make Ready the R1 mirror	<code>ready r1</code>	Sets the source (R1) device to be RDF Ready to its local host.
Make Ready the R2 mirror	<code>ready r2</code>	Sets the target (R2) device to be RDF Ready to its local host.
Make the R1 mirror Not Ready	<code>not_ready r1</code>	Sets the source (R1) device to be RDF Not Ready to its local host.
Make the R2 mirror Not Ready	<code>not_ready r2</code>	Sets the target (R2) device to be RDF Not Ready to its local host.

Table 2-3 Singular SRDF Control Operations (continued)

Control Operation	symrdf Action Arguments	Results
Merge the track tables of the R1 and R2 devices	merge	Merges the track tables between the source (R1) and the target (R2) side.
Enable consistency protection	enable	Enables consistency protection for SRDF/A capable devices.
Disable consistency protection	disable	Disables consistency protection for SRDF/A capable devices.
Confirm R2 data copy	checkpoint	Confirms to the caller that data in the current SRDF/A cycle has been committed to the R2.

The singular SRDF control operations outlined in Table 2-3 are described in the upcoming pages of this section.

Suspend

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

The suspend control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName suspend
symrdf -cg CgName suspend
symrdf -f[file] FileName suspend
```

For example, to suspend the SRDF links between all the RDF pairs in device group *prod*, enter:

```
symrdf -g prod suspend
```

To suspend the SRDF links between one RDF pair, *DEV007*, in device group *prod*, enter:

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

To suspend the SRDF links (between the pairs) on a list of RDF pairs in device group *prod*, enter:

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

To invoke a suspend, the RDF pair(s) must already be in one of the following states:

- ◆ Synchronized
- ◆ R1 Updated

When the suspend has completed successfully, the devices will be suspended on the SRDF links and their link status set to Not Ready (NR).

Note: This operation will be rejected if any of the following occur:

- If the source has invalid local (R1) tracks
 - If any of the device pairs are in one of the following states and the `-force` option is **not** specified:
 - SyncInProgress state
 - UpdInProgress state
 - Invalid state
 - Split state and the link status is Write Disabled
 - Suspended state and the link status is Write Disabled
 - Device Domino mode
 - When the source has invalid remote (R2) tracks
 - When the target has invalid local (R2) tracks
 - When consistency is enabled and the `-force` option isn't specified
-

Resume

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

The resume control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName resume
symrdf -cg CgName resume
symrdf -f[file] FileName resume
```

For example, to resume the SRDF links between all the RDF pairs in device group `prod`, enter:

```
symrdf -g prod resume
```

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

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

Note: This operation will be rejected if a merge track table is needed but has not been executed unless the `-force` option is specified.

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

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

To invoke this operation, the RDF pair(s) must already be in the Suspended state.

Read/Write Enable R1 Mirror

The *read/write enable R1 mirror* action argument `write` enables the source (R1) devices to their local hosts.

The read/write enable R1 mirror control operation can be performed by 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
```

For example, to read/write enable all the source (R1) mirrors in all the RDF pairs in device group `prod`, enter:

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

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

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

To invoke this operation, the RDF device must be in either the Write Disabled or the Not Ready state at the source, and the pairs must already be in one of the following states:

- ◆ Synchronized
- ◆ SyncInProgress
- ◆ Suspended
- ◆ Partitioned while you are invoking this operation from the source side
- ◆ Invalid

Note: This operation will be rejected if any of the device pairs are in the Failed Over state without specifying the `-force` option.

Read/Write Enable R2 Mirror

The *read/write enable R2 mirror* action argument write enables the target (R2) devices to their local hosts.

The read/write enable R2 mirror control operation can be performed by 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
```

For example, to Read/Write Enable all the target (R2) mirrors in the RDF pairs in device group `prod`, enter:

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

To Read/Write Enable the target (R2) mirror in one RDF pair, `DEV007`, in device group `prod`, enter:

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

To Read/Write Enable the target (R2) mirror of a list of RDF pairs in device group `prod`, enter:

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

To invoke this operation, the RDF pair(s) must already be in one of the following states:

- ◆ Suspended and Write Disabled at the target on the SA or RA
- ◆ Suspended and Not Ready at the target on the SA or RA
- ◆ Partitioned while you are invoking this operation from the target side and the devices are Write Disabled or Not Ready at the target on the SA or RA

Write Disable R1 Mirror

The *write disable R1 mirror* action argument write disables the source (R1) devices to their local hosts.

The write disable R1 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName write_disable r1
symrdf -cg CgName write_disable r1
symrdf -f[file] FileName write_disable r1
```

For example, to Write Disable all the source (R1) mirrors in the RDF pairs in device group `prod`, enter:

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

To Write Disable the source (R1) mirror in the RDF pair, `DEV007`, in device group `prod`, enter:

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

To Write Disable the source (R1) mirror in a list of RDF pairs, (`DEV002`, `DEV003`, `DEV007`) in device group `prod`, enter:

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

This operation can be invoked from the source side if the RDF pair(s) are already in the Partitioned state and the device is Ready on the SA at the source.

Note: This operation will be rejected if the device pair(s) are in one the following states, and Ready on the SA at the source without specifying the `-force` option:

- Synchronized
 - SyncInProgress
 - Suspended
 - Split
 - Invalid
-

Write Disable R2 Mirror

The *write disable R2 mirror* action argument write disables the target (R2) devices to their local hosts.

The write disable R2 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName write_disable r2
symrdf -cg CgName write_disable r2
symrdf -f[file] FileName write_disable r2
```

For example, to write disable all the target (R2) mirrors in the RDF pairs in device group `prod`, enter:

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

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

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

To write disable the target (R2) mirror in a list of RDF pairs in device group prod, enter:

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

This operation can be invoked from the target side if the device pairs are already in the Partitioned state and are Ready on the RA at the target side.

Note: This operation will be rejected if any of the following occur:

- If the device pairs are in one of the following states:
 - Suspended
 - Synchronized
 - SyncInProgress
 - If the device pair(s) are in one of the following states without specifying the `-force` option:
 - Split
 - Failed Over
 - R1 Updated
 - R1 UpdInProgress
 - Invalid
-

Read/Write Disable R2 Mirror

The *read /write disable R2 mirror* action argument blocks both reads and writes to the target (R2) devices to their local host. This option enables a user to set a device to the Not Ready state on the R2 side by making the device Not Ready on the RA.

The read/write disable R2 mirror control operation can be performed by device group, composite group, or device file:

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

For example, to read/write disable all the target (R2) mirrors in the RDF pairs in a device group `prod`, enter:

```
symrdf -g prod rw_disable r2
```

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

```
symrdf -g prod rw_disable r2 DEV007
```

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

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

This operation can be invoked from the target side if the device pair(s) are already in the Partitioned state and are Ready on the RA at the target side. Invoking this operation enables a distinction between Enginuity setting a device to Not Ready and the API disabling it. This enables the user flexibility for planned data processing operations involving the SRDF link.

Note: This operation will be rejected if the device pair(s) are in one of the following states without specifying the `-force` option:

- Split
 - Failed over
 - R1 Updated
 - R1 UpdInProg
 - Invalid
-

Refresh R1 Mirror

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

The refresh R1 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName refresh r1
symrdf -cg CgName refresh r1
symrdf -f[file] FileName refresh r1
```

For example, to refresh all the source (R1) devices in all the RDF pairs in the device group `prod`, enter:

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

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

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

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

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

To invoke this operation, the RDF pair(s) must already be in one of the following states:

- ◆ Suspended and Write Disabled at the source
- ◆ Suspended and Not Ready at the source
- ◆ Failed Over with the `-force` option specified

Note: This operation will be rejected if the target has invalid local (R2) tracks.

Refresh R2 Mirror

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

The refresh R2 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName refresh r2
symrdf -cg CgName refresh r2
symrdf -f[file] FileName refresh r2
```

For example, to refresh the target (R2) devices in all the RDF pairs in device group `prod`, enter:

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

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

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

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

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

To invoke this operation, the RDF pairs at the source must already be Suspended, and one of the following:

- ◆ Write Disabled or Not Ready at the source
- ◆ Ready at the source with the `-force` option specified

Note: This operation will be rejected if the source has invalid local (R1) tracks.

Invalidate R1 Mirror

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

The invalidate R1 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName invalidate r1
symrdf -cg CgName invalidate r1
symrdf -f[file] FileName invalidate r1
```

For example, to invalidate the source (R1) devices in all the RDF pairs in device group `prod`, enter:

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

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

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

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

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

To invoke this operation, the RDF pairs at the source must already be Suspended and Write Disabled or Not Ready.

Note: This operation will be rejected if the target has invalid local (R2) tracks.

Invalidate R2 Mirror

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

The invalidate R2 mirror control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName invalidate r2
symrdf -cg CgName invalidate r2
symrdf -f[file] FileName invalidate r2
```

For example, to invalidate the target (R2) devices in all the RDF pairs in device group `prod`, enter:

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

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

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

To invoke this operation, the RDF pair(s) at the source must already be Suspended and Write Disabled or Not Ready.

Note: This operation will be rejected if the source has invalid local (R1) tracks.

Ready R1 Mirror

The *make R1 mirror Ready* action argument sets the source (R1) devices to be RDF Ready to their local hosts. This operation may only be needed after all SRDF links have been lost when running in the RDF domino mode.

The make R1 mirror ready control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName ready r1
symrdf -cg CgName ready r1
symrdf -f[file] FileName ready r1
```

For example, to make the source (R1) device Ready in all the RDF pairs in device group `prod`, enter:

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

To make the source (R1) device Ready in one RDF pair, `DEV007`, in device group `prod`, enter:

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

To make the source (R1) device Ready in a list of RDF pairs in device group `prod`, enter:

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

This action can be invoked in all RDF states, except when you are invoking the action from the target side and the device pairs are in the Partitioned state.

Ready R2 Mirror

The *make R2 mirror Ready* action argument sets the target (R2) devices to be RDF Ready to their local hosts.

The make R2 mirror ready control operation can be performed by device group, composite group, or device file:

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

For example, to make the target (R2) devices Ready in all the RDF pairs in device group `prod`, enter:

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

To make the target (R2) device Ready in one RDF pair, `DEV007`, in device group `prod`, enter:

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

This action can be invoked in all RDF states, except when you are invoking the action from the source side and the device pair(s) are in the Partitioned state.

Not Ready R1 Mirror

The *make R1 mirror Not Ready* action argument sets the source (R1) devices to be RDF Not Ready to their local hosts.

The make R1 mirror not ready control operation can be performed by device group, composite group, or device file:

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

For example, to make the source (R1) devices Not Ready in all the RDF pairs in device group `prod`, enter:

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

To make the source (R1) device Not Ready in one RDF pair, DEV007, in device group `prod`, enter:

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

To make the source (R1) device Not Ready in a list of RDF pairs, DEV007, in device group `prod`, enter:

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

To invoke this operation, the RDF pair(s) must be Write Disabled at the source and already be in one of the following states:

- ◆ Failed Over
- ◆ R1 Updated
- ◆ R1 UpdInProg
- ◆ Suspended
- ◆ Partitioned and you are invoking this action from the source side

Not Ready R2 Mirror

The *make R2 mirror Not Ready* action argument sets the target (R2) devices to be RDF Not Ready to their local hosts.

The make R2 mirror not ready control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName not_ready r2
symrdf -cg CgName not_ready r2
symrdf -f[file] FileName not_ready r2
```

For example, to make the target (R2) devices Not Ready in all the RDF pairs in device group `prod`, enter:

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

To make the target (R2) device in one RDF pair Not Ready, DEV007, in device group `prod`, enter:

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

To invoke this operation, the RDF pair(s) must be Write Disabled at the target and already in one of the following states:

- ◆ Synchronized
- ◆ SyncInProgress
- ◆ Suspended
- ◆ Partitioned and you are invoking the action from the target side
- ◆ Invalid

Merge Track Tables

The *merge track tables* action argument merges the track tables between the source (R1) and the target (R2) devices. This option allows for the comparison of track tables on RDF device pairs in a device group and may be used to compare the track tables between RDF device pairs that have been split and re-established.

The merge track tables control operation can be performed by device group, composite group, or device file:

```
symrdf -g DgName merge
symrdf -cg CgName merge
symrdf -f[file] FileName merge
```

For example, to merge the track tables of all the RDF pairs in device group `prod`, enter:

```
symrdf -g prod merge
```

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

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

To merge the track table of a list RDF pairs in device group `prod`, enter:

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

To invoke this operation, the RDF pair(s) must already be in one of the following states:

- ◆ Suspended and the force option is specified, or the device pair(s) are Write Disabled or Not Ready at the source side
- ◆ Failed Over and the force option is specified

Note: This operation is rejected if any of the following occur:

- The source has invalid local (R1) tracks and the target has invalid local (R2) tracks.
 - The source has invalid remote (R2) tracks and the target has invalid remote (R1) tracks.
 - The source has invalid local (R1) tracks and the device is Read/Write Enabled at the source.
 - The target has invalid remote (R1) tracks and the device is Read/Write Enabled at the source.
 - The source has invalid remote (R2) tracks and the device is Read/Write Enabled at the target.
 - The target has invalid local (R2) tracks and the device is Read/Write Enabled at the target.
 - The source or target has local and remote invalid tracks.
-

Enable Consistency Protection

The *enable* action enables consistency protection for SRDF/A-capable devices by device group or device list. If data cannot be copied from the R1 to the R2, all devices in the group will be made Not Ready on the link to preserve R2 data consistency.

To enable consistency protection for SRDF/A pairs in device group *prod*, enter:

```
symrdf -g prod enable
```

To enable consistency protection for SRDF/A pairs listed in device file *devfile1*, enter:

```
symrdf -file devfile1 enable
```

Note: To enable consistency protection for SRDF/A pairs listed in a composite group (*-cg*), refer to Chapter 3, *RDF Consistency Group Operations* on page 3-55.

Disable Consistency Protection

The *disable* action disables consistency protection for SRDF/A capable devices by device group or device list. If data cannot be copied from the R2 to the R1, then only the devices in the group experiencing problems will be made Not Ready on the link. The device state for any remaining devices in the group will remain the same.

To disable consistency protection for SRDF/A pairs in device group `prod`, enter:

```
symrdf -g prod disable
```

To disable consistency protection for SRDF/A pairs listed in device file `devfile1`, enter:

```
symrdf -file devfile1 disable
```

Confirm R2 Data Copy

The *checkpoint* action confirms to the caller that data written in the current SRDF/A cycle has been successfully committed to the R2. The option is only valid for SRDF/A capable devices participating in an active cycle. All supplied devices must be in the same SRDF/A session.

Optionally, you can target devices in a device group or device list:

```
symrdf -g DgName checkpoint  
symrdf -f[file] FileName checkpoint
```

for example, to confirm R2 data copy for SRDF/A-capable devices in device group `prod`, enter:

```
symrdf -g prod checkpoint
```

SRDF States

SRDF devices that are operational are always in some RDF pair state. The RDF pair state encompasses the SRDF state on the source (R1) side, the SRDF links status, and the RDF state on the target (R2) side.

When you invoke a control operation on an RDF pair, the RDF pair state may be changed. This depends on whether the RDF state of the source (R1) side, status of the SRDF link, or the RDF state of the target (R2) side has changed. The status of devices can change if their front-end or back-end director changes in the SRDF link.

Figure 2-9 shows the possible states of the SRDF link and the RDF devices.

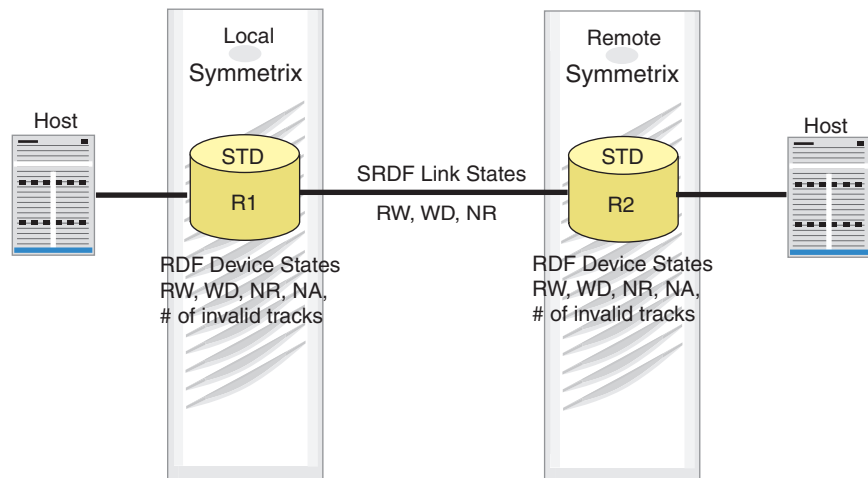


Figure 2-9 RDF Pair and Link States

SRDF Pair States

When you invoke a composite or singular control action on an SRDF pair, the SRDF pair state may be changed. This depends on whether the RDF state of the source (R1) side, status of the SRDF link, or the RDF state of the target (R2) side has changed. Before SRDF control operations can be successfully invoked, the SRDF pair state must be valid for that operation.

Table 2-4 provides a description of the various SRDF pair states.

Table 2-4 SRDF Pair States

State	Description
SyncInProg	A synchronization is currently in progress between the R1 and the R2. There are existing invalid tracks between the two pairs and the logical link between both sides of an RDF pair is up.
Synchronized	The R1 and the R2 are currently in a synchronized state. The same content exists on the R2 as the R1. There are no invalid tracks between the two pairs.
Split	The R1 and the R2 are currently Ready to their hosts, but the link is Not Ready or Write Disabled.
Failed Over	The R1 is currently Not Ready or Write Disabled and operations been failed over to the R2.
R1 Updated	The R1 is currently Not Ready or Write Disabled to the host, there are no local invalid tracks on the R1 side, and the link is Ready or Write Disabled.
R1 UpdInProg	The R1 is currently Not Ready or Write Disabled to the host, there are invalid local (R1) tracks on the source side, and the link is Ready or Write Disabled.
Suspended	The RDF links have been suspended and are Not Ready or Write Disabled. If the R1 is Ready while the links are suspended, any I/O will accumulate as invalid tracks owed to the R2.
Partitioned	The SYMAPI is currently unable to communicate through the corresponding RDF path to the remote Symmetrix. Partitioned may apply to devices within an RA group. For example, if SYMAPI is unable to communicate to a remote Symmetrix via an RA group, devices in that RA group will be marked as being in the Partitioned state.
Mixed	A composite SYMAPI device group RDF pair state. There exists different SRDF pair states within a device group.
Invalid	This is the default state when no other SRDF state applies. The combination of R1, R2, and RDF link states and statuses do not match any other pair state. This state may occur if there is a problem at the disk director level.
Consistent	The R2 SRDF/A capable devices are in a consistent state. Consistent state signifies the normal state of operation for device pairs operating in asynchronous mode.

Table 2-5 shows the various RDF pair states that result from the possible combined states of the source (R1) devices, the SRDF link, and the target (R2) devices.

The following legend provides details about elements in Table 2-5:

Table Legend

Not Ready: Disabled for both reads and writes

Ready: Enabled for both reads and writes

WD: Write Disabled

Table 2-5 SRDF States for the RDF Devices and Link

RDF pair State	Source (R1) RDF Status	RDF Link Status	Target (R2) RDF Status	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 ^a
R1 UpdInProg	Not Ready or WD	Ready (RW) or WD	Ready (RW)	> 0 ^a
Split	Ready (RW)	Not Ready or WD	Ready (RW)	—
SyncInProg	Ready (RW)	Ready (RW)	Not Ready or WD	> 0
Suspended	Any status ^b	Not Ready or WD	Not Ready or Write Disabled	—
Partitioned ^c	Any status	Not Ready	Not Available	—
Partitioned ^d	Not Available	Not Ready	Any status	—
Mixed	*e	*e	*e	—
Invalid ^f	Any status ^g	Any status	Any status	—
Consistent	Ready (RW) ^f	Ready (RW)	Not Ready or WD	0

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. Mixed state is seen only with `symdg show` to indicate that there are different device states in the group.

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

g. The combination of source RDF, SRDF links, and target RDF statuses do not match any other RDF state; therefore, the RDF state is considered Invalid.

RDF Operations and Applicable States

When RDF control operations are initiated, the RDF state of the device pair(s) is checked. If the device pair is not in a legal RDF state to initiate the control operation, the control will be blocked unless the `-force` option is used to force the pair to a specified RDF state.

For example, to initiate a failover on all the RDF pairs currently in the Split state in the `prod` group, enter:

```
symrdf -g prod -force failover
```

To initiate a failover on one RDF pair, DEV001, currently in the SyncInProg state in the `prod` group, enter:

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

Note: If devices are running in asynchronous mode (SRDF/A), control operations for a restore, update R1, and failback shall require the use of the `-force` option.

Table 2-6 describes which RDF control operations can be invoked for a given RDF state, noted by the check (✓). The `-force` option must be used to force a pair to a specified RDF state (where noted in Table 2-6 as ✓g).

Note: Additional table keys are presented in list form at the end of the table.

Table 2-6 RDF Control Operations and Applicable States

Control Operation	S y n c h r o n i z e d P r o g	S y n c h r o n i z e d	S p l i t	S u s p e n d e d	F a i l e d O v e r	P a r t i t i o n e d	R 1 U p d a t e d	R 1 U p d i n P r o g	I n v a l i d	C o n s i s t e n t
establish -full			✓	✓ ^a	✓ ^m				✓ ^j	
split	✓ ⁿ	✓ ⁱ		✓			✓			✓
establish			✓	✓					✓ ^j	

Table 2-6 RDF Control Operations and Applicable States (continued)

Control Operation	S ync In Prog	S yn ch ron ized	S p lit	S us p ended	F a iled O ver	P a r t i t i o n ed	R 1 U p d a t ed	R 1 U p d In Prog	I n v a l id	C o n s i s t e n t
restore			✓	✓ ^e					✓ ^k	
restore -full			✓	✓ ^a					✓ ^l	
suspend	✓ ⁿ	✓		✓ ^{g,o}	✓ ^{g,o}		✓	✓ ⁿ	✓ ^g	✓
resume				✓						
failover	✓ ⁿ	✓	✓ ^g	✓		✓ ^h	✓	✓ ⁿ	✓ ^g	✓
failback			✓	✓ ^a	✓	✓ ^{f,g}	✓	✓		
update				✓ ^a	✓		✓			
ready r1		✓	✓	✓	✓	✓ ^f	✓	✓	✓	✓
ready r2	✓ ^q	✓	✓	✓	✓	✓ ^h	✓	✓	✓ ^q	
not_ready r1				✓ ^p	✓	✓ ^p	✓	✓	✓	
not_ready r2	✓ ^q	✓		✓		✓ ^c			✓ ^q	
invalidate r1				✓ ^a						
refresh r1				✓ ^a	✓ ^g					
invalidate r2				✓ ^a						
refresh r2				✓ ^{a,g}						
merge				✓ ^e	✓ ^g					
rw_enable r1	✓ ^a	✓ ^a		✓ ^a	✓ ^g	✓ ^{a,f}			✓ ^a	✓
write_disable r1	✓ ^{a,g}	✓ ^{b,g}	✓ ^{b,g}	✓ ^{b,g}		✓ ^{b,g}			✓ ^{b,g}	✓

Table 2-6 RDF Control Operations and Applicable States (continued)

Control Operation	S y n c h r o n i z e d	S y n c h r o n i z e d	S p l i t	S u s p e n d e d	F a i l e d O v e r	P a r t i t i o n e d	R 1 U p d a t e d	R 1 U p d a t e d P r o g	I n v a l i d	C o n s i s t e n t
rw_enable r2				✓ _c		✓ _{c,h}				
write_disable r2			✓ _g		✓ _g	✓ _{d,h}	✓ _g	✓ _g	✓ _{d,g}	
rw_disable r2	✓ _q	✓		✓		✓ _c			✓ _{gq}	
swap				✓ _p	✓		✓			
deletepair			✓	✓	✓					
enable	✓	✓	✓	✓	✓	✓ _h	✓	✓		✓
disable	✓	✓	✓	✓	✓	✓ _h	✓	✓		✓
checkpoint	✓	✓								✓

- a. SA Write Disabled or Not Ready on the source side.
- b. SA Ready on the source side.
- c. SA or RA Write Disabled or Not Ready on the target side.
- d. RA Ready on the target side.
- e. Use Force or SA Write Disabled or Not Ready on the source side.
- f. Host application run while connected to the source side.
- g. The `-force` option must be used.
- h. Host application run while connected to the target side.
- i. The `-force` option must be used when in Adaptive Copy mode.
- j. R1 and R2 are Not Ready but the link is Ready and there are no local or remote invalid tracks on the source or target.
- k. R1 and R2 are Not Ready but the link is Ready and there are no remote invalid tracks on the source.
- l. R1 and R2 are Not Ready but the link is Ready and there are no local or remote invalid tracks on the source.
- m. R1 is not visible to any host.
- n. The `-symforce` option can be used.

- o. Write Disabled on the SRDF link.
- p. Write Disabled on the R1 side.
- q. Not allowed when in SRDF/Asynchronous mode.

Setting SRDF Modes

You can use the SYMCLI `symrdf` command to modify SRDF modes on remotely mirrored standard devices in a device group, composite group, or device file. The SYMCLI syntax to set modes is as follows:

```
symrdf -g DgName set mode  
symrdf -cg CgName set mode  
symrdf -f[file] FileName set mode
```

Using the `symrdf set mode` command, you can modify the modes, which are described below.

Synchronous

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

For example, to set the remotely mirrored pair in the `prod` group to the Synchronous mode, enter:

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

This state ensures that the source (R1) and target (R2) devices contain identical data.

Semi-Synchronous

In the semi-synchronous mode, the Symmetrix array containing the source (R1) device informs the host of successful completion of the write operation when it receives the data. The RDF (RA) director transfers each write to the target (R2) device as the RDF links become available. The Symmetrix array containing the target (R2) device checks and acknowledges receipt of each write.

If a new write is started for a source (R1) device before the previous write has completed to the target (R2) device, the Symmetrix array containing the source (R1) device temporarily disconnects from the I/O bus until the previous write operation is completed and acknowledged from the remote Symmetrix array, and then reconnects to the I/O bus and continues processing.

For example, to set all the remotely mirrored pairs in the device group `prod` to the semi-synchronous mode, enter:

```
symrdf -g prod set mode semi
```

Asynchronous

In the SRDF/Asynchronous mode (SRDF/A), the Symmetrix array provides a consistent point-in-time image on the target (R2) device, which is a short period of time behind the source (R1) device. Managed in sessions, SRDF/A transfers data in predefined timed cycles or *delta sets* to ensure that data at the remote (R2) site is *dependent write consistent*. This mode requires an SRDF/A license.

The Symmetrix 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 can be 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 Symmetrix *ordered write processing*, which transfers the changed tracks over the link only once within any single cycle.

For example, to set the remotely mirrored pair in the `prod` group to the asynchronous mode, enter:

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

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 and TimeFinder Clone operations. Refer to Appendix A for details.

For additional information on operating in asynchronous mode, refer to Chapter 3, *SRDF/Asynchronous Operations* on page 3-9.

Domino Effect On

The device domino effect mode ensures that the data on the source (R1) and target (R2) devices are always in sync. The Symmetrix array will force the source (R1) device to a Not Ready state and respond “*intervention required/unit not ready*” to the host whenever it detects one side in a remotely mirrored pair is unavailable, or all link failures have occurred and the host tries to access the device.

For example, to turn the device domino effect on for the `prod` device group, enter:

```
symrdf -g prod set domino on
```

After the problem has been corrected, the *Not Ready* device must be made *Ready* again to the host using the `symrdf ready` command.

For example, to make all the source (R1) side devices Ready in the device group `prod`, enter:

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

If the failed device or links are still not available when the SRDF device is made Ready, the device becomes Not Ready again when the device is accessed.

Note: The RDF consistency state for the pair must be disabled before allowing Domino mode to be enabled. Domino mode cannot be enabled for SRDF/A-capable devices.

Some important issues to consider:

- ◆ When the device domino effect is ON, you will not be able to use the `split` or `suspend` control operation because it would cause the devices to become Not Ready.
- ◆ All SRDF links will still fail regardless if link domino is enabled when all RDF R1 devices become Not Ready.

Domino Effect Off

Under normal operating conditions (domino effect not enabled), a remotely mirrored device will continue processing I/Os with its host, even when an SRDF device or link failure occurs. New data written to the source (R1) or target (R2) device while its pair is unavailable or link paths are out of service are marked for later transfer. When link paths are re-established or the device becomes available, resynchronization begins between the source (R1) and target (R2) devices.

For example, to turn the domino effect off for the device group `prod`, enter:

```
symrdf -g prod set domino off
```

Adaptive Copy Write Pending

When you set the SRDF mode to adaptive copy write pending mode, the Symmetrix array acknowledges all writes to the source (R1) device as if it was a local device. The 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.

For example, to turn on the adaptive copy write pending mode for the device group `prod`, enter:

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

To turn off adaptive copy write pending mode for the device group `prod`, enter:

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

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

Adaptive Copy - Disk

The adaptive copy disk mode is designed for situations requiring the transfer of large amounts of data without loss of performance. Because the Symmetrix array cannot fully guard against data loss should a failure occur, it is recommended that you use this mode temporarily to transfer the bulk of your data to target (R2) devices, and then switch to a full SRDF mode (synchronous or semi-synchronous) or adaptive copy-write pending mode (if you can tolerate some lack of synchronization between the remotely mirrored pairs) to ensure full data protection.

When you set the SRDF mode to adaptive copy disk, the Symmetrix 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.

For example, to turn on the adaptive copy disk mode for the `prod` group, enter:

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

To turn the adaptive copy disk mode off for the `prod` group, enter:

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

This attribute also has a user-configurable **skew** (maximum number of invalid tracks threshold), that when exceeded, causes the remotely mirrored device to operate in the predetermined SRDF state (synchronous or semi-synchronous) when this mode is in effect. As soon as the number of invalid tracks drops for a device below this value, the remotely mirrored pair reverts back to the adaptive copy write pending mode.

**Adaptive Copy
Change Skew**

This attribute is used to modify the adaptive copy **skew** threshold. When the skew threshold is exceeded, the remotely mirrored pair operates in the predetermined SRDF state (synchronous or semi-synchronous). As soon as the number of invalid tracks drop below this value, the remotely mirrored pair reverts back to the adaptive copy mode.

The skew value is configured at the device level and may be set to a value between 0 and 65,534 tracks. For devices larger than a 2 GB capacity drive, a value of 65,535 can be specified to target all the tracks of any given drive.

For example, to change the adaptive copy skew value to all tracks of device BCV023 of group `prod`, enter:

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

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

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

Not Ready if Invalid

This attribute is used to set the R2 Not Ready if there are invalid tracks. Invalid tracks could be either remote invalids on the source (R1) side or target (R2) invalids on the source (R1) side. Invalid tracks occur when the user enables the target (R2) for Read/Write status. If there are invalid tracks, the device RDF status is set to Not Ready. Set the `nr_if_invalid` argument to on or off.

For example, to set the target (R2) side of the SRDF configuration to the Not Ready state if there are invalid tracks for device BCV023 of group `prod`, enter:

```
symrdf -g prod set nr_if_invalid on BCV023
```

Dynamic SRDF Pair Operations

SRDF device pairing was previously limited to the static SRDF pairs set at Symmetrix configuration time. Since Enginuity 5568, dynamic RDF enables the creation and deletion of SRDF pairs while the Symmetrix array is in operation. Once established, the new SRDF pairs can be synchronized and managed in the same way as configured SRDF pairs.

Note: Since Enginuity 5670, a dynamic R1 device can be converted to a concurrent RDF device by dynamically adding a second remote mirror. Refer to *Creating Dynamic Concurrent SRDF Pairs* on page 2-63.

Requirements

Dynamic SRDF pairing requires an SRDF licence with Symmetrix Enginuity 5568 or higher. The dynamic RDF configuration state of the Symmetrix array must be enabled and the devices must be designated as dynamic `RDF_capable` devices. With the Symmetrix array in operation, the configuration change commands can be used to change these settings. Refer to the *EMC Solutions Enabler Symmetrix Configuration Change CLI Product Guide*.

Note: For Enginuity 5567 systems, dynamic RDF operations are limited to swap only. Refer to *Dynamic R1/R2 Swap* on page 2-67.

Note: Dynamic SRDF pair operations including `createpair`, `deletepair` and `swap` are supported for SRDF/A mode using Enginuity 5671.

Display RDF Capable Devices

To display devices that have been configured as dynamic `RDF_capable`, you can use the `symdev list` command with the `-dynamic` option as follows:

```
symdev list -dynamic [-R1] [-R2]
```

From the displayed list, determine which dynamic devices on the source Symmetrix array you would like to pair with dynamic devices on the target Symmetrix array. If `-R1` or `-R2` is not specified, all devices that are RDF-capable will be displayed.

Create Device File

Once you have identified the dynamic RDF devices, a separate device list of the new pairs must be created in a text file. The device file syntax contains two columns (R1 and R2 designated devices). Each SRDF pair must be listed on a separate line in the device file. The `-type` option is used to indicate whether the device in the first column is an R1 or an R2. The devices listed in the first column associate with the specified Symmetrix ID.

In the following example, local devices are listed in the first column and remote devices are listed in the second column:

```
# Device File
010A  00B7
010F  00BF
0106  00C5
```

For an example of the command line syntax used to execute the `symrdf createpair` command using the `-file` and `-type` options, refer to *Creating Dynamic SRDF Pairs with Invalidate* on page 2-61.

Creating Dynamic Pairs with a Device File

To dynamically create a number of SRDF paired devices, use the following syntax:

```
symrdf -file Filename -sid SymmID -RDFG GrpNum
        [-h] [-v|noecho] [-noprompt] [-force]
        [-bypass] [-i Interval] [-c count]
        createpair [-g DgName -type [RDF1|RDF2]]
        -invalidate [R1|R2]
        -establish
        -restore [-remote]
```

As you create pairs from the device list, you select one of three possible and subsequent control operations:

- ◆ The `invalidate` option (`-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 RDF pairs are created.
- ◆ The `establish` option (`-establish`) begins copying data to the invalidated target(s), synchronizing the dynamic RDF pairs in the device file once the RDF pairs are created.
- ◆ The `restore` option (`- restore`) begins copying data to the source device(s), synchronizing the dynamic RDF pairs in the device file once the RDF pairs are created.

Note: For concurrent RDF operations, you can also apply the `-remote` option to failback, restore, and update R1 devices in a concurrent pair configuration by converting the R1 devices in the device file to concurrent devices. Refer to Chapter 3, *The Remote Option for Restore, Update, Failback* on page 3-27.

Creating Dynamic SRDF Pairs with Invalidate

The following example dynamically creates a number of SRDF paired devices and invalidates the targets.

```
symrdf createpair -file devicefile -sid 55 -rdfg 1\  
-type RDF1 -invalidate r2
```

Here the `symrdf createpair` command creates new RDF pairs from the list of devices in the device file. For this example, a file called `devicefile` identifies the first-column devices that are in Symmetrix 55 as R1 type devices. The R2 devices are invalidated with this create pair example and the RDF pairs become members of RDFG 1. Upon execution of this command, this pairing information will be added to the SYMAPI database file on the host.

You can now establish the RDF pairs in the list, which copies data to the invalidated target devices. Then you can query the action to check the progress of the establish operation. Once synchronized, you can perform various SRDF operations on members of the device file.

```
symrdf -file devicefile establish -sid 55 -rdfg 1  
symrdf -file devicefile query -sid 55 -rdfg 1
```

Creating Dynamic SRDF Pairs with Establish

Optionally, you can include the establish operation in the `createpair` command line by replacing the `-invalidate r2` option described earlier with the `-establish` option, where the default copy path is R1 to R2 for all the device pairs in the list as follows:

```
symrdf createpair -file devicefile -sid 55 -rdfg 1\  
-type RDF1 -establish
```

Once the RDF device pairs are created, the establish operation begins copying data to the targets, synchronizing the device pairs listed in the device file.

Note: For the `createpair -establish` option, the R2 may be set to Read/Write Disabled (Not Ready) if `SYMAPI_RDF_RW_DISABLE_R2=ENABLE` is set in the options file. For more information, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

Creating Dynamic SRDF Pairs for a Restore

You can perform a restore operation to copy data back to the R1 source device(s) by including the `-restore` option in the `createpair` command line as follows:

```
symrdf createpair -file devicefile -sid 55 -rdfg 1\
                -type RDF1 -restore
```

Once the RDF device pairs are created, the restore operation begins copying data to the source device(s), synchronizing the dynamic RDF device pairs listed in the device file.

Createpair Restrictions

The `symrdf createpair` operation will be rejected if any of the following apply:

- ◆ The device is in one of the following BCV pair states: Synchronized, SyncInProgress, Restored, RestoreInProgress, SplitInProgress.
- ◆ The device is the source or target of a TimeFinder/Snap operation.
- ◆ There is a background BCV split operation in progress.
- ◆ Devices are in the backend Not Ready state.
- ◆ There is an optimizer swap in progress on a device.
- ◆ The emulation type is not the same (i.e., AS/400 has specific pairing rules).
- ◆ There are existing local invalid tracks on either the local or remote device.
- ◆ Devices are not Write Disabled on the SCSI adapter and the `-invalidate` option was used.
- ◆ The SRDF/A session is active.
- ◆ The RDF group is in asynchronous mode and the devices being added are not the same RDF type R1 or R2.
- ◆ The RDF group is in asynchronous mode and the `-establish` or `-restore` option is selected.

- ◆ The RDF group is enabled for RDF consistency.
- ◆ 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
 - PPRC mainframe system

Creating Dynamic Concurrent SRDF Pairs

Since Enginuity 5670, you can dynamically create concurrent RDF pairs using the `symrdf createpair` command. This feature allows a second remote mirror to be dynamically added by converting a dynamic R1 device to a concurrent RDF device. Two remote mirrors are supported for any dynamic R1 device.

Note: Beginning with Enginuity 5671, SRDF/Asynchronous devices are supported for dynamic concurrent RDF.

For information explaining a concurrent SRDF configuration, refer to *Concurrent RDF Operations* on page 3-23.

The following rules apply when creating a dynamic concurrent SRDF pair:

- ◆ Remote BCVs that have been designated as dynamic RDF devices are not supported.
- ◆ The concurrent device grouping for the two remote second mirrors must be assigned to different RA groups.
- ◆ The concurrent dynamic RDF, dynamic RDF, and concurrent RDF states must be enabled in your Symmetrix configuration.
- ◆ Only one RA group can be enabled for SRDF/A (asynchronous) mode.
- ◆ The remote second mirror cannot be enabled for consistency protection.
- ◆ With the `-restore` selection, the `-remote` option is required if the link status for the first created remote mirror is Read/Write.

To dynamically create a second remote mirror using the `symrdf createpair` command, you must create two separate device files: One file containing the first set of R1/R2 device pairs, and a second device file listing the same R1 device paired with a different remote R2 device.

To create a concurrent mirror for the `createpair` example previously described, you can use the following example:

```
symrdf createpair -file devicefile2 -sid 55 -rdfg 2\
                -type RDF1 -invalidate R1|R2
```

Here the `symrdf createpair` command creates new RDF pairs from the list of devices in a second device file. For this example, a file called `devicefile2` identifies the first-column devices that are in Symmetrix 55 as R1 type devices. The second-column devices become the second remote mirror devices. The RDF device pairs will become members of RDF group 2. Also, you can execute a restore/remote operation to restore the standard R2 type devices:

```
symrdf createpair -file devicefile2 -sid 55 -rdfg 2\
                -type RDF1 -restore -remote
```

Note: The concurrent mirror devices must belong to a separate RA group than those defined in the first device file pairing.

Delete Dynamic SRDF Pairs

Dynamically created SRDF pairs can be deleted by using the `symrdf deletepair` command. This command cancels the dynamic SRDF pairing by removing the pairing information from the Symmetrix array.

Note: You must suspend the RDF links using the `symrdf suspend` command before performing the `symrdf deletepair` command.

For example:

```
symrdf -file devicefile suspend -sid 55 -rdfg 2
symrdf deletepair -file devicefile -sid 55 -rdfg 2
```

Here the RDF link has been suspended for the devices listed in the device file and that reside on source Symmetrix 000125600055. The `-rdfg 2` is the RDF group number by which the pairs communicate.

After execution of the `symrdf deletepair` command, the dynamic SRDF pairs have been canceled, the pairing information has been removed from the Symmetrix array and SYMAPI database, and the devices have been changed to non-RDF devices (except when an RDF concurrent pair exists).

Note: The `symrdf deletepair` operation will be rejected if any of the following apply:

- The device is in one of the following BCV pair states: Synchronized, SyncInProgress, Restored, RestoreInProgress, SplitInProgress.
- The device is the source or target of a TimeFinder/Snap operation.
- There is a background BCV split operation in progress.
- Devices are in the back end Not Ready state.
- There is an optimizer swap in progress on a device.
- The emulation type is not the same (i.e., AS/400 has specific pairing rules).
- The SRDF pairs are in one of the RDF in progress states.
- There are existing local invalid tracks on either the local or remote device.
- There are remote invalid tracks and the `-force` option was not specified.
- RDF Consistency is enabled.
- The links are not suspended.

The `symrdf deletepair` command is allowable in all of the following states, where the links are Not Ready status:

- ◆ Suspended
- ◆ Split
- ◆ Failed Over

Control of Dynamic Pairs by Device Group

SRDF allows you to perform subsequent control operations on previously created dynamic SRDF pairs by referencing a device group (`-g`) instead of specifying the device file. To implement dynamic SRDF control by device group:

1. List your device pairings in a device file, and then create dynamic SRDF pairs, applying the `-g GroupName` option to the command line. This adds the devices listed in the device file to a device group (*NewGrp*) as follows:

```
symrdf createpair -file devicefile -sid 55 -rdfig 2 \
    -type rdf1 -invalidate r2 -g NewGrp
```

2. Perform SRDF control operations on the dynamic SRDF pairs within the device group. For example, establish the group:

```
symrdf -g NewGrp establish
```

The `symrdf createpair` command created dynamic SRDF pairs from the device file and added the pairs to a device group called *NewGrp*. The `symrdf establish` command then performed an establish operation on the dynamic SRDF pairs in the device group *NewGrp*. All SRDF commands for these dynamic pairs can now be executed within the context of the device group, including the `symrdf deletepair` command.

For example:

```
symrdf deletepair -g NewGrp
```

The `symrdf deletepair` command, changes the devices within the group to non-RDF devices and changes the SYMAPI device group to a regular device group (except when an RDF concurrent pair exists).

Note: The RDF links must be suspended prior to deleting SRDF pairs using this functionality.

If additional devices were added to the device group prior to the `symrdf deletepair` command being used, those added devices would also be changed to non-RDF devices, and the device group to a regular device group, only if the added devices contained within it were dynamic devices. If the device group contained both RDF and non-RDF devices, the device group would be changed to an Invalid state.

Dynamic R1/R2 Swap

You can swap the RDF personality of the RDF device designations of a specified device group if you have one of the following:

- ◆ An SRDF license and a Configuration Manager license with Enginuity 5x66 or 5267.
- ◆ An SRDF license and a Configuration Manager license with Enginuity 5567 or higher, and the devices to be swapped are not dynamic RDF-enabled in your Symmetrix configuration.
- ◆ An SRDF license with Enginuity 5567 or higher and dynamic RDF is enabled in your Symmetrix configuration.

With a dynamic swap, source R1 devices become target R2 devices and target R2 devices become source R1 devices.

Swaps using dynamic RDF, perform faster, but must be enabled in your Symmetrix configuration to use this feature.

Note: Dynamic swap is not supported:

- If both Enginuity 5567 and 5568 systems are mixed across your local and remote Symmetrix enterprise.
- For Enginuity 5669 and higher systems where the R2 device is larger than the R1 device.
- For concurrent RDF devices.

Since Enginuity 5568, the dynamic RDF configuration state of the Symmetrix must be enabled for the swap operation. Dynamic RDF-capable devices are configured as one of three types: RDF1 capable, RDF2 capable, or both. Dynamic R1/R2 Swap capability requires that the devices be configured as both to initiate a swap.

When swapping the RA group personalities that engage ESCON directors in a FarPoint connection, be aware that FarPoint buffer settings cannot be adjusted using `symconfigure`. If your FarPoint buffers are set to customized parameters other than default values, an EMC representative will need to be called to adjust the buffer settings after the swap has taken place.

Display RDF Swap-Capable Devices

To display RDF devices that have been configured as dynamic RDF-capable, you can use the `symrdf list` command with the `-dynamic` option as follows:

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

If no option is specified, all RDF devices that are RDF-capable will be displayed. Use the `-R1` option to display all dynamic RDF-capable devices that are configured as capable of becoming R1. Use the `-R2` option to display all dynamic RDF-capable devices that are configured as capable of becoming R2.

To display a list of dynamic RDF-capable devices that are configured as capable of becoming R1 or R2, use the `-both` option as follows:

```
symrdf list -dynamic -both
```

From the displayed list, determine which dynamic devices you want to swap.

Swap RDF Devices

To perform an R1/R2 swap, use the following form:

```
symrdf -g DgName [-h] [-force] [-bcv|-all] [-v|-noecho]
          [-bypass] [-i Interval] [-c Count] [-noprompt]
swap [-refresh R1|R2]
```

The `-bcv|all` option lets you target just the BCV-associated devices (`-bcv`) for the swap action in the RDF. Use `-all` to target both BCV and standard devices. Use nothing to target just the standard devices.

The `-refresh` option marks the source R1 device(s) or the target R2 device(s) to refresh from the remote mirror.

R1/R2 Swap Example

The following example swaps the R1 designation of the associated BCV RDF1 devices within device group `ProdGrpB`. It also, marks to refresh any modified data on the current R1 side of these BCVs from their R2 mirrors, enter:

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

Refresh Data Concerns

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 will mark these 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.

Data Status Concerns Swapping the R1/R2 designation of the RDF devices can impact the state of your stored data as shown in Table 2-7.

Table 2-7 RDF Device Data Status for a Swap

RDF Side With Data	Swap Operation, Refresh Target Selection
RDF1	<code>-refresh R1</code> —The R2 device holds the valid copy and the R1 device's invalid tracks will be updated using the R2 data.
RDF1	<code>-refresh R2</code> —The R1 device holds the valid copy and the R2 device's invalid tracks will be updated using the R1 data.
RDF2	<code>-refresh R1</code> —The R2 device holds the valid copy and the R1 device's invalid tracks will be updated using the R2 data.
RDF2	<code>-refresh R2</code> —The R1 device holds the valid copy and the R2 device's invalid tracks will be updated using the R1 data.

Legal States Before a Swap Operation The current states of the various devices involved in the SRDF swap must be considered *before* executing a swap action. Table 2-8 lists which states are legal for this operation.

Table 2-8 RDF Device States Before Swap Operation

RDF 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

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 devices.

Disable SYMAPI Behavior Parameter In the options file, behavior parameter `SYMAPI_CTRL_OF_NONVISIBLE_DEVS` must be enabled if the devices are not mapped to the user host.

SRDF/A Dynamic Swap Beginning with Engenuity 5671, dynamic swap can be performed on devices in SRDF/Asynchronous mode.

Note: However, dynamic swap cannot be performed for SRDF/A devices that are enabled for consistency protection or if the SRDF/A session is actively copying.

Dynamic Failover Establish

RDF dynamic devices can be quickly failed over, swapped, and then re-established all within a single command-line operation.

The `symrdf failover -establish` command can be used as a composite operation on dynamic RDF devices to quickly perform the following single control operations together:

- ◆ Fail over from the R1 to the R2.
- ◆ Dynamic RDF swap.
- ◆ Incremental establish on the swapped RDF pairs.

Note: Support for dynamic failover establish functionality using the `symrdf failover -establish` command is available since Enginuity 5567. Functionality requires that dynamic devices be both RDF1- and RDF2-capable.

Restrictions

Certain current restrictions that apply for dynamic failover establish are as follows:

- ◆ RDF devices configured as RAID-S are not supported.
- ◆ RDF devices set for device domino mode or configured using the `-link_domino` option are not supported.
- ◆ RDF devices within a composite group that has been enabled for RDF consistency are not supported.
- ◆ RDF device pairs where the R2 device is larger than the R1 device are not supported.
- ◆ Concurrent RDF configurations are not supported.
- ◆ Configurations using FarPoint are not supported.

Command Functionality

When the `symrdf failover -establish` command is issued, the RDF devices in the group will perform all of the necessary steps in the following order. First, the devices will be failed over, making the R2 devices in the group read/write enabled to their local hosts. Refer to *Failover* on page 2-21 for a detailed explanation of a failover operation.

Note: If there are invalid tracks encountered during the establish step of the failover operation, a merge track table operation will automatically be performed before the failover establish operation continues. This extra step may stretch the execution time of the operation.

After the failover operation has completed, the RDF pairs will swap personalities (the R1 devices become R2 devices and the R2 devices become R1 devices). Refer to *Dynamic R1/R2 Swap* on page 2-67 for a detailed explanation and restrictions that apply when performing a dynamic swap operation.

Once the devices have been dynamically swapped, an incremental establish operation is initiated and the devices become immediately available on the link. Refer to *Incremental Establish* on page 2-11 for details.

symrdf -g Command Line Options

With the `symrdf -g` command, you can perform RDF control operations on RDF device(s) in a device group. The control operations (arguments) have options that allow flexibility in controlling the RDF pairs.

Table 2-9 lists the `symrdf` control operations type arguments and options available when operating on RDF device(s) of a specified device group.

Table 2-9 symrdf -g Control Arguments and Possible Options

	Possible Options											
Argument Action	-force -sym force	-by pass	-bcv	-brbcv -rbcv	-all	-rdtg	-re mote	-imme diate	-i/-c	-v	-no echo	-no prompt
deletepair	X	X				X			X	X	X	X
disable	X	X	X	X	X				X	X	X	X
enable	X	X	X	X	X				X	X	X	X
establish [-full]	X	X	X	X	X	X			X	X	X	X
failback	X	X	X	X	X	X	X		X	X	X	X
failover [-establish]	X	X	X	X	X	X		X	X	X	X	X
invalidate r1 r2	X	X	X	X	X	X			X	X	X	X
merge	X	X	X	X	X	X			X	X	X	X
not_ready r1 r2	X	X	X	X	X	X			X	X	X	X
ready r1 r2	X	X	X	X	X	X			X	X	X	X
refresh r1 r2	X	X	X	X	X	X			X	X	X	X
restore [-full]	X	X	X	X	X	X	X		X	X	X	X
resume	X	X	X	X	X	X			X	X	X	X
rw_disable r2	X	X	X	X	X	X			X	X	X	X
rw_enable r1 r2	X	X	X	X	X	X			X	X	X	X

Table 2-9 symrdf -g Control Arguments and Possible Options (continued)

Argument Action	Possible Options											
	-force -sym force	-by pass	-bcv	-brbcv -rbcv	-all	-rdfg	-re mote	-imme diate	-i/-c	-v	-no echo	-no prompt
set mode domino skew nr	X	X	X	X	X	X			X	X		X
split	X	X	X	X	X	X		X	X	X	X	X
suspend	X	X	X	X	X	X		X	X	X	X	X
swap ^a	X	X	X		X	X			X	X	X	X
update [-until]	X	X	X	X	X	X	X		X	X	X	X
write_disable r1 r2	X	X	X	X	X	X			X	X	X	X

a. Another possible option is `-refresh R1|R2` (refer to *Swap RDF Devices* on page 2-68).

Note: To enable the `-symforce` option for use, a behavior parameter called `SYMAPI_ALLOW_RDF_SYMFORCE` in the options file must be set to TRUE.

Table 2-10 lists the `symrdf` view action type arguments and options available when viewing RDF device(s) of a specified device group.

Table 2-10 symrdf -g View Arguments and Possible Options

Options	Argument Action		
	query	verify	checkpoint
-all	X	X	X
-bcv	X	X	X
-consistent		X	
-failedover		X	
-i, -c	X	X	X
-offline	X	X	X
-partitioned		X	
-rbcv, -brbcv	X	X	X

Table 2-10 **symrdf -g View Arguments and Possible Options (continued)**

Options	Argument Action		
	query	verify	checkpoint
-rdfa	X		
-rdfg	X	X	X
-split		X	
-susp_offline		X	
-suspended, -enabled		X	
-synchronized		X	
-syncinprog		X	
-updated		X	
-updateinprog		X	
-valid		X	

Table 2-11 lists the `symrdf` view action type arguments and options available when viewing RDF device(s).

Table 2-11 **symrdf -g View Arguments and Possible Options**

Options	Argument Action	
	list	ping
-bcv	X	
-concurrent	X	
-consistency	X	
-dynamic	X	
-i, -c	X	X
-nobcv	X	
-offline	X	
-R1, -R2	X	
-rdf		X

Table 2-11 symrdf -g View Arguments and Possible Options (continued)

Options	Argument Action	
	list	ping
-rdfa	X	
-rdfg	X	
-resv	X	
-sid	X	X
-v	X	

The following sections provide brief descriptions of all the various SRDF control options identified in these tables.

Targeting All Devices

The all (-all) option targets the SRDF action at all devices in the device group, which includes standard RDF devices and any BCV RDF devices that are locally associated with the device or composite group.

Targeting BCV Devices

The BCV (-bcv) option allows the SRDF control operation to target the specified BCV device(s) that are associated with the device or composite group and are configured as RDF BCV devices. By default, only the SRDF standard devices are affected by the SRDF control operations.

The no BCV (-nobcv) option allows the SRDF control operation to target specified devices, not including BCV devices.

The BCV remote BCV (-brbcv) option allows you to target the SRDF action at the specified remotely associated RDF (Hop 2) BCV devices which can be paired with the remote mirrors of the local BCV devices.

The remote BCV (-rbcv) option allows you to target the SRDF action at the specified remotely associated RDF (Hop 2) BCV devices, which can be paired with the remote mirrors of the local standard devices.

Bypassing Locks

The bypass (`-bypass`) option causes the SRDF control operation to bypass existing Symmetrix exclusive locks.



CAUTION

Use the `-bypass` option ONLY if you are SURE that no other SRDF operation is in progress in either the local and/or remote Symmetrix arrays.

Listing Devices

To list RDF devices that are visible to your host, or RDF devices configured on a given Symmetrix array, use the `symrdf list` command. This command can be used with the following options to confine the list to specific devices.

Concurrent Devices

The concurrent (`-concurrent`) option confines the list action to just the devices that are configured as concurrent RDF.

Dynamic RDF Devices

The dynamic (`-dynamic`) option confines the list action to just the devices that are configured as dynamic RDF.

SRDF/A Capable Devices

The SRDF/A-capable device option (`-rdfa`) allows you to list devices that are SRDF/A-capable.

Note: Beginning with Engenuity 5671, all devices are SRDF/A-capable and the command will display all devices.

RDF Group Devices

The RDFG (`-rdfg`) option applies a Symmetrix RDF (RA) group number to the command to restrict the list to just the RDF devices of an RDF group. When used with control, verify, and query actions, this option targets a specific RDF group number or all groups when devices are configured RDF concurrent. The RDF group syntax for all the composite commands is `-rdfg nn|All`.

R1 or R2 Devices

The R1 or R2 (`-R1 | -R2`) options allows you to list only the devices that are RDF1 types (`-R1`) or RDF2 types (`-R2`).

Devices with SCSI Reservations

The SCSI reservations (`-resv`) option allows you to list RDF devices that have SCSI reservations.

For example, to list all the RDF devices in Symmetrix array 333 that have SCSI reservations, enter:

```
symrdf -sid 333 -resv list
```

Consistency State

The consistency state (`-consistency`) option allows you to list the RDF consistency state when you are listing RDF devices.

For example, to show the consistency state in the list of all the RDF devices in Symmetrix array 333, enter:

```
symrdf -sid 333 -consistency list
```

BCV Devices

The BCV (`-bcv`) option allows you to list only the bcv devices and the `-nobcv` option allows you to exclude the bcv devices, listing only the standard SRDF devices.

Running Repetitive Commands

The interval (`-i`) option executes a command in repeat intervals to display or to attempt to acquire an exclusive lock on the Symmetrix host database, the local Symmetrix, and the remote Symmetrix units. The default interval is 10 seconds. The minimum interval is 5 seconds.

The count (`-c`) option counts the number of times to display or to attempt acquiring exclusive locks on the Symmetrix host database, the local Symmetrix array, and the remote Symmetrix arrays.

If the (`-c`) option is not specified and an interval (`-i`) is specified, the program loops continuously to produce infinite redisplay, or until the RDF control or set operation starts.

To query device group `prod` every 10 seconds for 1 minute, enter:

```
symrdf -g prod -i 10 -c 6 query
```

Forcing a Rejected State

The force (`-force`) option allows you to perform control operations on SRDF devices when SRDF device(s) are not in the expected state for the control operation.

Using the `-force` option, the control operation will be attempted, regardless of the state of the SRDF devices, if it is legal to do so according to the rules in Table 2-6 on page 2-50.

For example, if one SRDF device in an RDF pair is in the Suspended state, and the other SRDF device is in the Synchronized state, to split the RDF pair, `DEV007`, in the device group `prod`, enter:

```
symrdf -g prod -force split DEV007
```

Forcing a Rejected State with Symforce



The `symforce` option (`-symforce`) requests that the Symmetrix array force the operation to occur that overrides instances where they are normally rejected.

CAUTION

Use care when applying this option as data could be lost or corrupted.

Note: To enable the `-symforce` option for RDF use, a behavior parameter called `SYMAPI_ALLOW_RDF_SYMFORCE` in the `options` file must be set to `TRUE`.

With `-symforce`, a split command will execute on an RDF pair, even when they are in a sync in progress state. During the execution of an establish or restore command, `-symforce` will inhibit the verification of valid tracks on the device at the source.

Getting Help

The help (`-h`) option allows you receive brief online help for the SYMCLI command.

Setting No Echo Display

The no echo (`-noecho`) option suppresses the display of information which results from an RDF control operation.

Setting No Prompt Confirmation

The no prompt (`-noprompt`) option suppresses the message asking you to confirm an RDF control operation.

Obtaining Information from the SYMAPI Database

The offline (`-offline`) option prevents accessing the Symmetrix array to update the database. The `symrdf` command uses information previously gathered from the Symmetrix array and held in the Symmetrix host database as opposed to interrogating the Symmetrix array directly. The offline option can alternatively be set by assigning the environment variable `SYMCLI_OFFLINE` to 1.

Remote Data Copying

The remote (`-remote`) option requests a remote data copy with the `failback`, `restore`, and `update` actions. When the concurrent link is Ready, data will also be copied to the concurrent RDF mirror. For these actions to execute, use this option or suspend the concurrent link.

Targeting a Symmetrix

The Symmetrix ID (`-sid`) option allows you to specify the Symmetrix array, which the command references.

Verifying Device States

You can verify the RDF pair state by using the `symrdf verify` command. This command can be used with the following options to verify specific device pair states.

Enabled Consistency State

The enabled state option (`-enabled`) verifies whether the RDF device pair(s) are in the Enabled Consistency state. Used in companion with the `-suspended` option.

Consistent State

The consistent state (`-consistent`) option verifies whether the R2 mirror of SRDF/A (capable) device pairs are in the R2 consistent pair state.

Failed Over State

The failed over state option (`-failedover`) verifies whether the RDF device pair(s) are in the Failed Over state.

Partitioned State

The partitioned state option (`-partitioned`) verifies whether the RDF device pair(s) are in the Partitioned state.

Split State

The split state option (`-split`) verifies whether the RDF device pair(s) are in the Split state.

Suspended State

The suspended state option (`-suspended`) verifies whether the RDF device pair(s) are in the Suspended state.

Suspended Offline State

The suspended offline state option (`-susp_offline`) verifies whether the RDF device pair(s) are in the Suspended state and the SRDF link is offline.

Synchronized State

The synchronized state option (`-synchronized`) verifies whether the RDF device pair(s) are in the Synchronized state. This is the verify default action.

To verify whether the RDF device pair(s) are currently in the process of synchronizing (SyncInProgress state), use the `-syncinprog` option.

Updated State

The updated state (`-updated`) option verifies whether the RDF pair(s) are in the Updated state.

For example, to verify whether the RDF device pair, `DEV007`, in the device group `prod` is in the Updated state, enter:

```
symrdf -g prod -updated verify DEV007
```

Update In Progress State

The update in progress state option (`-updateinprog`) verifies whether the RDF device pair(s) are in the UpdateInProg state.

Valid State

The valid state (`-valid`) option verifies whether the RDF pair(s) are in a valid RDF pair state (all RDF pair states except Invalid).

For example, to verify whether all the RDF devices in the device group `prod` are in one of the valid RDF pair states, enter:

```
symrdf -g prod -valid verify
```

Setting the Number of Invalid Track Updates

The until (`-until`) option when used with the `update` argument checks the number of invalid tracks that are allowed to build up from the active R2 local I/O before another update (R2 to R1) copy is retriggered. The update sequence loops until the invalid track count is less than the number specified for the `-until` value. Refer to *Continuous R1 Updates* on page 2-27 for more information.

Displaying Command Status

The verbose (`-v`) option displays status and progress information as it executes the desired operation.

Dropping the SRDF/A Session

The drop session immediately (`-immediate`) option causes the *failover*, *split*, or *suspend* operation to drop the SRDF/A session immediately. Refer to Chapter 3, *Using the Immediate Option* on page 3-17 for more information.

symrdf -cg Command Line Options

With the `symrdf -cg` command, you can perform RDF control operations on RDF device(s) in a composite group. The control operations (arguments) have options that allow flexibility in controlling the RDF pairs.

Table 2-12 lists the `symrdf` control operations type arguments and options available when operating on RDF device(s) of a specified composite group.

Table 2-12 symrdf -cg Control Arguments and Possible Options

	Possible Options							
Argument Action	-force -sym force	-by pass	-bcv	-rdfg	-i/-c	-v	-no echo	-no prompt
establish [-full]	X	X	X	X	X	X	X	X
failback	X	X	X	X	X	X	X	X
failover [-establish]	X	X	X	X	X	X	X	X
invalidate r1 r2	X	X	X	X	X	X	X	X
merge	X	X	X	X	X	X	X	X
not_ready r1 r2	X	X	X	X	X	X	X	X
ready r1 r2	X	X	X	X	X	X	X	X
refresh r1 r2	X	X	X	X	X	X	X	X
restore [-full]	X	X	X	X	X	X	X	X
resume	X	X	X	X	X	X	X	X
rw_enable r1 r2	X	X	X	X	X	X	X	X
set mode domino skew nr		X	X	X	X	X		X
split	X	X	X	X	X	X	X	X

Table 2-12 **symrdf -cg Control Arguments and Possible Options (continued)**

	Possible Options							
Argument Action	-force -sym force	-by pass	-bcv	-rdfg	-i/-c	-v	-no echo	-no prompt
suspend	X	X	X	X	X	X	X	X
update [-until]	X	X	X	X	X	X	X	X
write_disable r1 r2	X	X	X	X	X	X	X	X

Note: To enable the `-symforce` option for use, a behavior parameter called `SYMAPI_ALLOW_RDF_SYMFORCE` in the `options` file must be set to `TRUE`.

Table 2-13 lists the `symrdf` view action type arguments and options available when viewing RDF device(s) of a specified composite group.

Table 2-13 **symrdf -cg View Arguments and Possible Options**

	Argument Action		
Options	query	verify	checkpoint
-bcv	X	X	
-consistent		X	
-cg_consistent		X	
-failedover		X	
-i, -c	X	X	X
-offline	X	X	X
-partitioned		X	
-split		X	
-susp_offline		X	
-suspended, -enabled		X	
-synchronized		X	

Table 2-13 symrdf -cg View Arguments and Possible Options (continued)

Options	Argument Action		
	query	verify	checkpoint
-syncinprog		X	
-updated		X	
-updateinprog		X	
-valid		X	

For a brief description of all the various options in these tables, refer to *Targeting All Devices* on page 2-75 where the option descriptions begin.

symrdf -file Command Line Options

With the `symrdf -file` command, you can perform RDF control operations on RDF device(s) listed in the specified device file. The control operations (arguments) have options which allow flexibility in controlling the RDF pairs.

Device File

The device file (`-file`) option directs the specified operation in the `symrdf` command to a device file. The device file contains device pairs (*SymDevnames*) listing a pair for each line. Device files can include comment lines that begin with the pound sign(#). The following example illustrates the file format, which specifies three device pairs:

```
00A1 0103
00A2 0104
#00A3 0105 (To be reinstalled later)
00B1 0106
```

When you use this option, you must specify a target Symmetrix ID or set environmental variable `SYMCLI_SID`. These options allow you to operate on Symmetrix arrays and remote BCV pairs beyond the first SRDF multi-hop.

Table 2-14 lists the `symrdf` control operations type arguments and options available when operating on RDF device(s) of a specified device file.

Table 2-14 symrdf -file Control Arguments and Possible Options

Argument Action	Possible Options										
	-sid	-type	-force -sym force	-by pass	-rd fg	-re mote	-imme diate	-i/-c	-v	-no echo	-no prompt
createpair [-g Name] -invalidate -establish -restore	X	X	X	X	X	X		X	X	X	X
deletepair	X		X	X	X			X	X	X	X
disable	X		X	X	X			X	X	X	X
enable	X		X	X	X			X	X	X	X
establish [-full]	X		X	X	X			X	X	X	X
failback	X		X	X	X	X		X	X	X	X

Table 2-14 symrdf -file Control Arguments and Possible Options (continued)

Argument Action	Possible Options										
	-sid	-type	-force -sym force	-by pass	-rdfg	-re mote	-imme diate	-i/-c	-v	-no echo	-no prompt
failover [-establish]	X		X	X	X		X	X	X	X	X
invalidate r1 r2	X		X	X	X			X	X	X	X
merge	X		X	X	X			X	X	X	X
not_ready r1 r2	X		X	X	X			X	X	X	X
ready r1 r2	X		X	X	X			X	X	X	X
refresh r1 r2	X		X	X	X			X	X	X	X
restore [-full]	X		X	X	X	X		X	X	X	X
resume	X		X	X	X			X	X	X	X
rw_enable r1 r2	X		X	X	X			X	X	X	X
rw_disable r1 r2	X		X	X	X			X	X	X	X
set mode domino skew	X			X	X			X	X		X
split	X		X	X	X		X	X	X	X	X
suspend	X		X	X	X		X	X	X	X	X
swap ^a	X		X	X	X			X	X	X	X
write_disable r1 r2	X		X	X	X			X	X	X	X

a. Another possible option is `-refresh R1|R2` (refer to *Swap RDF Devices* on page 2-68).

To enable the `-symforce` option for use, a behavior parameter called `SYMAPI_ALLOW_RDF_SYMFORCE` in the options file must be set to `TRUE`.

Table 2-15 lists the `symrdf` view action type arguments and options available when viewing RDF device(s) listed in a specified device file.

Table 2-15 symrdf -file View Arguments and Possible Options

Options	Argument Action		
	query	verify	checkpoint
-consistent		X	
-failedover		X	
-i, -c	X	X	X
-offline	X	X	X
-partitioned		X	
-rdfa	X		
-rdfg	X	X	X
-sid	X	X	X
-split		X	
-susp_offline		X	
-suspended, -enabled		X	
-synchronized		X	
-syncinprog		X	
-updated		X	
-updateinprog		X	

For a brief description of all the various options in these tables, refer to *Targeting All Devices* on page 2-75 where the option descriptions begin.

Various Remote Operations

This chapter discusses operation, management, and strategies of various possible Symmetrix Remote Data Facility (SRDF) configurations and how to perform special operations.

The terms SRDF and RDF are used throughout this chapter and refer to the Symmetrix Remote Data Facility.

- ◆ Multi-Hop Operations.....3-2
- ◆ RDF Group Topologies in an SRDF.....3-7
- ◆ SRDF/Asynchronous Operations3-9
- ◆ Dynamic SRDF Group Operations.....3-19
- ◆ Concurrent RDF Operations.....3-23
- ◆ TimeFinder Consistent Splits Across RDF3-28
- ◆ SRDF/Automated Replication Operations.....3-31
- ◆ Relative Symmetrix Manager Functions3-54
- ◆ RDF Consistency Group Operations.....3-55

Multi-Hop Operations

Various compounded remote configurations can be managed by your host using both the TimeFinder and SRDF components of SYMCLI.

As shown in Figure 3-1 on page 3-4, you can have multiple sites (for example, remote Sites B and C) on SRDF links to remotely mirror a local Symmetrix array at Site A. Remote Site B, functioning as a remote mirror to the standard devices at Site A, is most typical. You then can have a third site on an SRDF link (remote Site C) to remotely mirror just the BCV devices in the Symmetrix array at Site A.

Multi-Hop SRDF Sites

You can also multi-hop to a second-level SRDF where Remote Site D functions as a remote mirror to the standard devices of Site A and Remote Site E remotely mirroring Site A's BCV.

Command `symrdf` manages the RDF pairs within the SRDF link while `symmir` manages the BCV pairs within any one site.

System-Wide Device Groups

Before you begin applying any `symmir` operations, you must be working with an existing group of RDF devices. To create a device group containing STD and BCV RDF1 devices, enter:

```
symdg create prod -type RDF1
symld -g prod add dev 0001 -sid 344402 DEV001
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
```

At this point, all these devices must be established with the `symmir` and `symrdf` commands.

The following set of examples illustrate how various `symmir` and `symrdf` commands might be applied to split operations throughout a complex remote configuration such as one shown in Figure 3-1 on page 3-4.

System-Wide Splits

Before you begin applying any `symmir` and `symrdf` operations, you must be working with an established group of RDF devices. To split the BCV pair within Site A, enter:

```
symmir -g prod split
```

To split RDF pairs at Site B from host-connected Site A, enter:

```
symrdf -g prod split
```

To split the BCV pairs within Site B, enter:

```
symmir -g prod -rdf split
```

To split BCV RDF pairs at Site C from host-connected Site A, enter:

```
symrdf -g prod -bcv split
```

To split the BCV pairs within Site C, enter:

```
symmir -g prod -rdf -bcv split
```

To split BCV RDF pairs at Site D from host standard-associated Site B, enter:

```
symrdf -g prod -rbcv split
```

To split the BCV pairs within Site D, enter:

```
symmir -f dfile -sid 0014 split  
or  
symmir -g prod -rrbcv split
```

To split BCV RDF pairs at Site E from host BCV-associated Site C, enter:

```
symrdf -g prod -brbcv split
```

To split the BCV pairs within Site E (hop 2), enter:

```
symmir -f dfile -sid 0015 split
```

Other operations for these remote sites such as establish and restore apply and execute in the same manner.

For more detail information about `symmir`, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

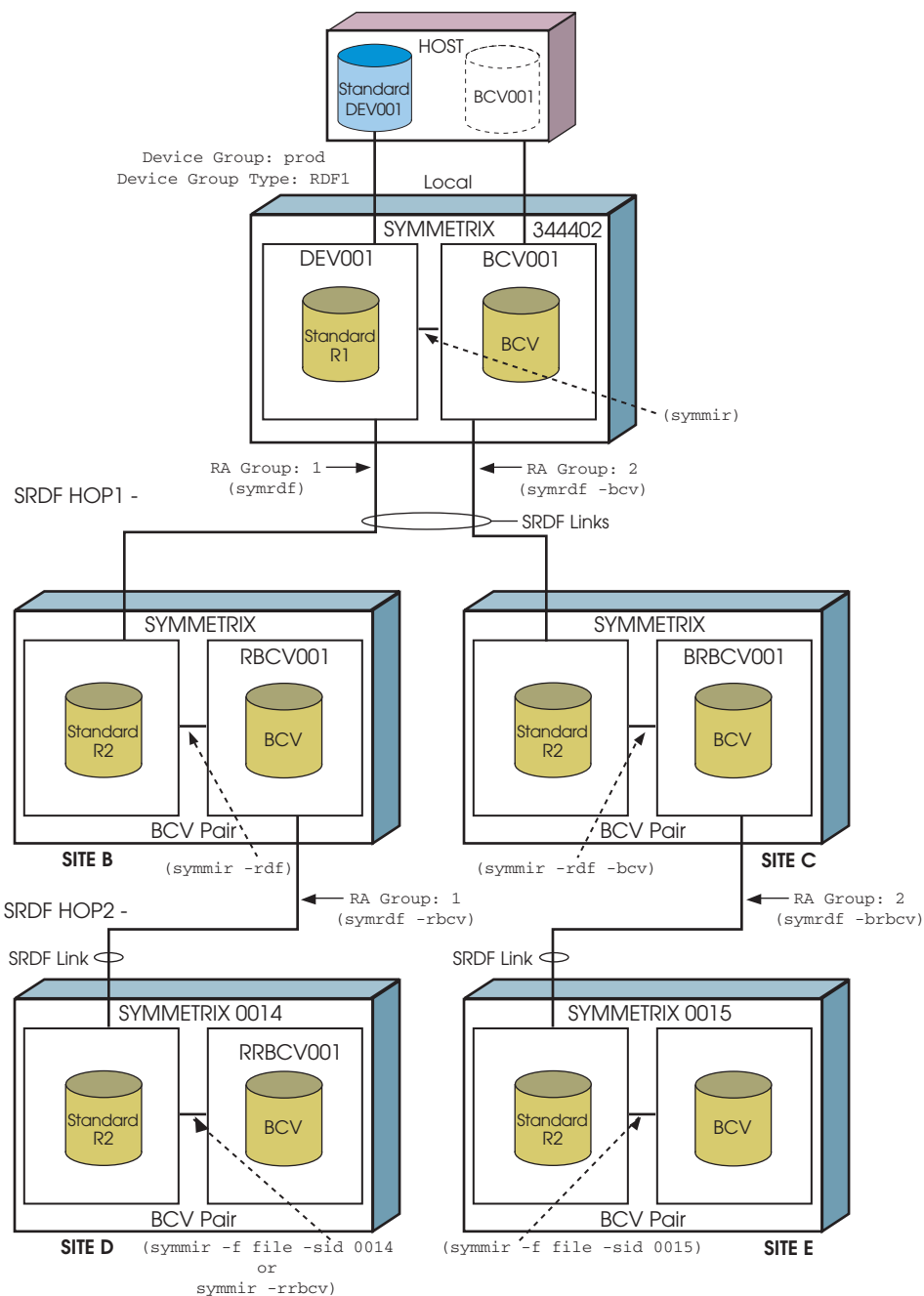


Figure 3-1 Various Remote Configurations

Targeting Commands to Various Multi-Hop Devices and Links

This section describes the command application of targeting the various devices and links in complex multi-hop RDF environments.

Table 3-1 shows a sequence of command steps for some basic control operations, which touch every device and SRDF link in a complex multi-hop configuration. This table works with and is illustrated by Figure 3-2 on page 3-6. The following numbering of commands directly associates with the bubble numbers shown in the figure.

Table 3-1 Remote Multi-Hop SRDF Commands

Step	CLI Control Operation	Description
1	<code>symrdf -g <> establish</code>	Creates the standard associated hop 1 copy.
2	<code>symmir -g <> split -rdf</code>	Splits the standard associated hop 1 BCV device pair.
3	<code>symrdf -g <> establish -rbcv</code>	Creates the standard associated hop 2 copy.
4	<code>symrdf -g <> restore -rbcv</code>	Restores the standard associated hop 1 BCV with the hop 2 copy.
5	<code>symmir -g <> restore -rdf</code>	Restores the standard associated hop 1 copy with the hop 1 BCV.
6	<code>symrdf -g <> restore</code>	Restores the standard device with the hop 1 copy.
7	<code>symmir -g <> split</code>	Splits the standard/BCV pair.
8	<code>symrdf -g <> establish -bcv</code>	Creates the BCV associated hop 1 remote copy.
9	<code>symmir -g <> split -rdf -bcv</code>	Splits the BCV associated hop 1 device pair.
10	<code>symrdf -g <> establish -brbcv</code>	Creates the BCV associated hop 2 copy.
11	<code>symrdf -g <> restore -brbcv</code>	Restores the BCV associated hop 1 BCV with the hop 2 copy.
12	<code>symmir -g <> restore -rdf -bcv</code>	Restores the standard device associated hop 1 copy with the hop 1 BCV.
13	<code>symrdf -g <> restore -bcv</code>	Restores the BCV device with the hop 1 copy.
14	<code>symmir -g <> restore</code>	Restores the standard device with the BCV copy.
15	<code>symmir -f <> -sid 056 establish</code> or <code>symmir -g <> -rrbcv establish</code>	Creates the BCV associated hop 2 BCV copy.

Table 3-1 Remote Multi-Hop SRDF Commands (continued)

Step	CLI Control Operation	Description
16	<code>symmir -f <> -sid 056 split</code> or <code>symmir -g <> -rrbcv split</code>	Splits the BCV-associated hop 2 device pair.

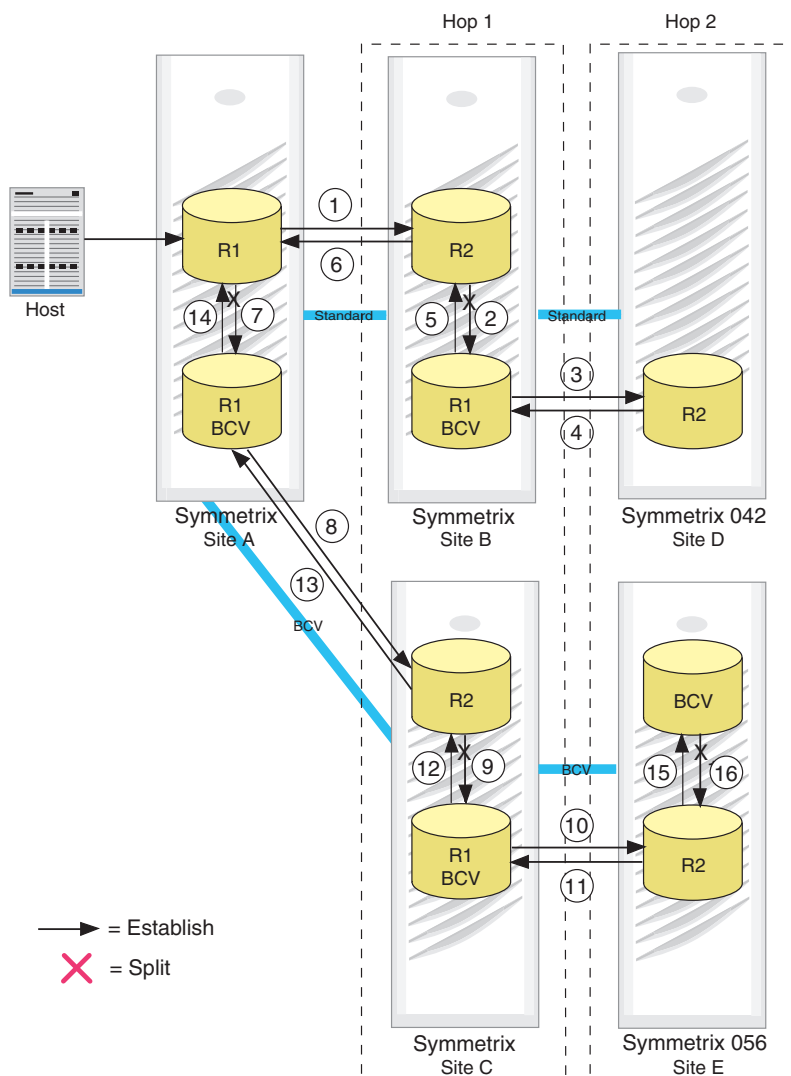


Figure 3-2 Remote Multi-Hop SRDF Configurations

RDF Group Topologies in an SRDF

There are two types of topologies for RDF (RA) groups and connection between Symmetrix arrays:

- ◆ RDF Groups in a Point-to-Point SRDF Link
- ◆ RDF Groups in a Switched SRDF Link

RDF Groups in a Point-to-Point SRDF Link

As shown in Figure 3-3, RA adapters interface to an SRDF link between Symmetrix arrays at different sites. RDF groups represent an established connection and an RDF pair associated link between certain R1 and R2 devices. R1 devices are RDF1 types and R2 are RDF2 types.

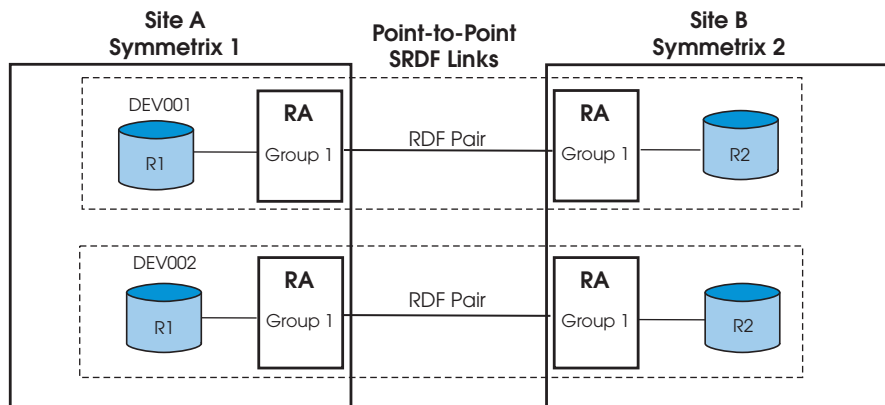


Figure 3-3 RDF Group Topology in a Point-to-Point SRDF Solution

RDF Groups in a Switched SRDF Link

As shown in Figure 3-4, RF adapters interface to an SRDF switched network (link) between Symmetrix arrays at different sites. RDF (RA) groups represent an established connection and an RDF pair associated switched link between certain R1 and R2 devices. R1 devices are RDF1 types and R2 are RDF2 types. In the figure, Groups 3 and 4, R2 device at Site B can be paired with Group 1 or 2, R1 device at Site A or Site B.

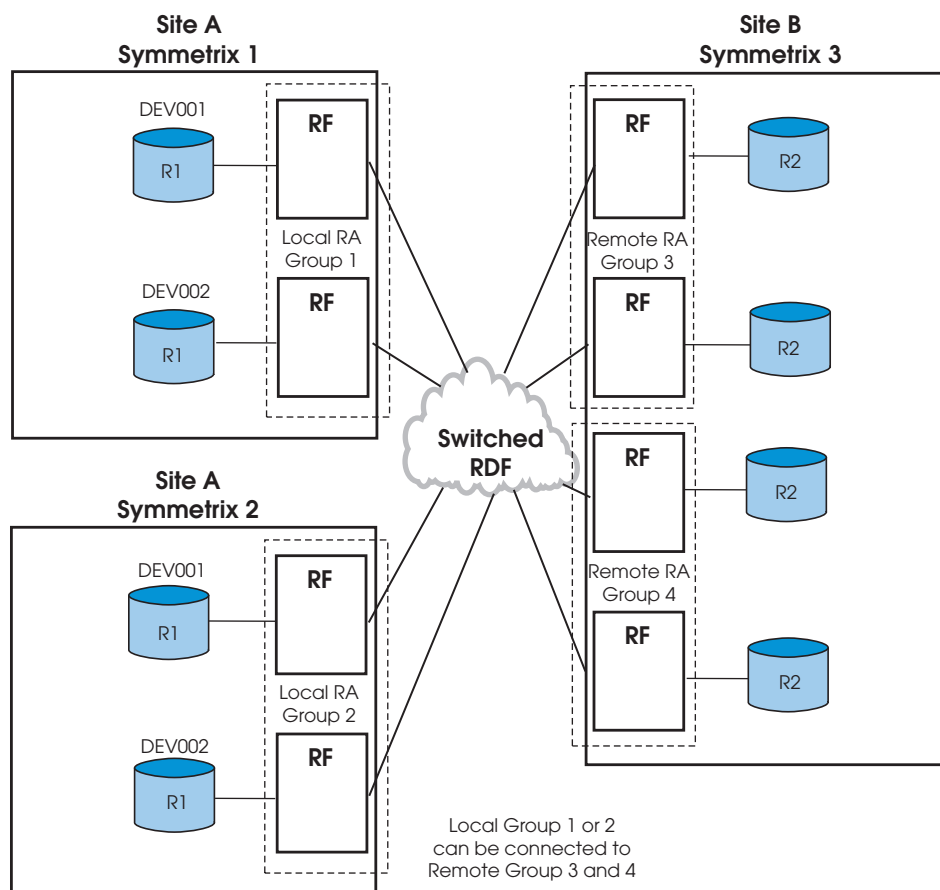


Figure 3-4 RDF Group Topology in a Switched RDF Solution

To view what local and remote Symmetrix arrays, their RF directors and RDF groups are connected via the open RDF switch fabric, enter:

```
symcfg list -RA ALL -switched
```

SRDF/Asynchronous Operations

Since Enginuity Version 5670, Symmetrix arrays support SRDF/Asynchronous (SRDF/A) mode for RDF devices. Asynchronous mode provides a point-in-time image on the target (R2) device, which is only slightly behind the source (R1) device. SRDF/A session data is transferred to the remote Symmetrix array in predefined timed cycles or *delta sets*, which minimizes the redundancy of same track changes being transferred over the link. This functionality requires an SRDF/A license.

SRDF/A provides a long-distance replication solution with minimal impact on performance that particularly preserves data consistency with the database. This level of protection is intended for backup environments that always need a restartable copy of data at the R2 site. In the event of a disaster at the R1 site or if RDF links are lost during data transfer, a partial delta set of data can be discarded, preserving consistency on the R2 with a maximum data loss of two SRDF/A cycles or less.

Note: For a description of each of the various SRDF modes, refer to *Setting SRDF Modes* on page 2-54.

SRDF/A Benefits and Features

SRDF/Asynchronous mode provides the following benefits and features:

- ◆ Provides lower operational cost for long-distance data replication with database consistency.
- ◆ Promotes efficient link utilization resulting in lower link bandwidth.
- ◆ Maintains a consistent point-in-time image on the R2 devices at all times.
- ◆ Supports all current SRDF topologies, including point-to-point and switched fabric.
- ◆ Requires no additional hardware, such as switches or routers.
- ◆ Has the ability to operate at any given distance without adding response time to the R1 host.
- ◆ Supports all hosts and data emulation types supported by the Symmetrix array (such as FBA, CKD, AS/400).

- ◆ Minimizes the impact imposed on the back-end DA directors.
- ◆ Provides a performance response time equivalent to writing to local non-SRDF devices.
- ◆ Allows restore, failover, and failback capability between the R1 and the R2 sites.
- ◆ Dynamic RDF pair operations including the `createpair`, `deletepair` and `swap` commands are supported for SRDF/A-capable devices. Functionality requires Enginuity 5671. Refer to Chapter 2, *Dynamic SRDF Pair Operations* on page 2-59 for additional information on dynamic pairing.
- ◆ Concurrent RDF devices that are SRDF/A-capable are supported. Functionality requires Enginuity 5671. Only one RDF group in a concurrent configuration can be run in asynchronous mode at one time, the other RDF group must be in a mode other than asynchronous. Refer to *Concurrent RDF Operations* on page 3-23 for additional information on concurrent configurations.
- ◆ Multiple SRDF/A sessions are allowed per Symmetrix array, with all 64 RDF groups being SRDF/A-capable. Functionality requires Enginuity 5671.
- ◆ SRDF/A-capable devices can be added to composite groups and be enabled for database consistency protection using Multi Session Consistency (MSC). Refer to *RDF Consistency Group Operations* on page 3-55 for additional information on SRDF/A consistency protection.
- ◆ Mode transition from asynchronous to synchronous ensuring database consistency for data on the R2 side is available for devices managed by device group. Functionality requires Enginuity 5671. Refer to *Mode Transition to Synchronous* on page 3-18.
- ◆ Capability to dynamically change some SRDF/A parameter settings through using the `symconfigure` command is available. Configurable parameters include: maximum cache utilization, maximum host throttle time, minimum cycle time, and SRDF/A group priority. Functionality requires Enginuity 5671. Refer to the *Symmetrix Configuration Change CLI Product Guide* for details on setting these parameters.

SRDF/A Restrictions

There are certain current restrictions and limitations for running SRDF/A-capable devices in asynchronous mode. The following is a list of current restrictions that apply:

General

- ◆ An SRDF/A license is required to access this functionality.
- ◆ All SRDF/A-capable devices running in asynchronous mode must be managed together by device group, composite group, or device list in an SRDF/A session.
- ◆ If there are tracks owed from the R2 to the R1, it is not recommended to switch to asynchronous mode. Although, the force option can be applied. The force option is required if there are tracks owed to the R1 device when attempting to make SRDF/A-capable devices in asynchronous mode Ready on the link.
- ◆ SRDF/A-capable devices that are enabled for consistency group protection must be disabled before attempting to change the mode from asynchronous.
- ◆ Asynchronous mode is currently limited to single-hop configurations using Enginuity 5670.
- ◆ Symmetrix RDF Automated Replication (SRDF/AR) control operations are currently **not** supported for SRDF/A-capable devices running in asynchronous mode.

Device Group

The following restrictions currently apply to device groups for SRDF/A-capable devices:

- ◆ Existing RDF1 and RDF2 device groups can be used to control SRDF/A-capable devices, but all devices of a certain type (i.e., standard, BCV, RBCV, or BRBCV) must be either SRDF/A-capable or non-SRDF/A-capable.
- ◆ All SRDF/A-capable devices in the group must be members in the same SRDF/A session within any device type.

Composite Group

The following restrictions currently apply to composite groups for SRDF/A-capable devices.

- ◆ All Devices within the group must be in asynchronous mode (or non-Asynchronous mode).
- ◆ Only standard devices can be managed in asynchronous mode.

Snapshots and Clones

Symmetrix arrays employing either TimeFinder/Snap or TimeFinder Clone operations affect whether RDF devices are allowed to be set in asynchronous mode. Refer to Appendix A, *TimeFinder/Snap and Clone State Reference* for a description of the TimeFinder/Snap and TimeFinder Clone pair states and setting RDF devices to asynchronous mode. Also, for SRDF/A-capable devices operating in asynchronous mode, certain Snap and Clone operations will not be allowed.

Note: For a list of TimeFinder/Snap or Clone operations not supported with asynchronous mode, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

Setting SRDF/Asynchronous Mode

To set remotely mirrored pairs in the device group `prod` to the asynchronous mode, enter:

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

Optionally, you can set asynchronous mode for devices in a composite group or device file:

```
symrdf -cg CgName set mode async
symrdf -f[file] FileName set mode async
```

A device status check is performed on all TimeFinder snapshot and Clone device pairs in the group before the operation is allowed. For details, refer to the current restrictions in the previous section, *Snapshots and Clones*, and Appendix A.

Note: Refer to *Setting SRDF Modes* on page 2-54 for a description of each of the various SRDF modes.

SRDF/A Session Monitoring

An SRDF/A session consists of a group of devices that has been set to Asynchronous mode. The SRDF/A session can be monitored through using the `symstat` command. Refer to the *EMC Solutions Enabler Base Management CLI Product Guide* for instructions on using the `symstat` command to obtain various Symmetrix performance statistics for monitoring an SRDF/A session.

SRDF/A Ordered-Write Processing

Different from traditional ordered-write processing, the Symmetrix array implements asynchronous mode host writes from the source to the target using predetermined timed cycles (called delta sets). Each delta set contains groups of I/Os for processing. Using cycles of operation, SRDF/A transfers sets of data, one cycle at a time between the R1 and the R2. If the same track is written to more than one time within an active set, SRDF/A will send the update over the link only once. This approach lowers the link bandwidth as compared with other ordered write-processing approaches, which transfer each write separately. Refer to Figure 3-5 for a depiction of SRDF/A delta sets.

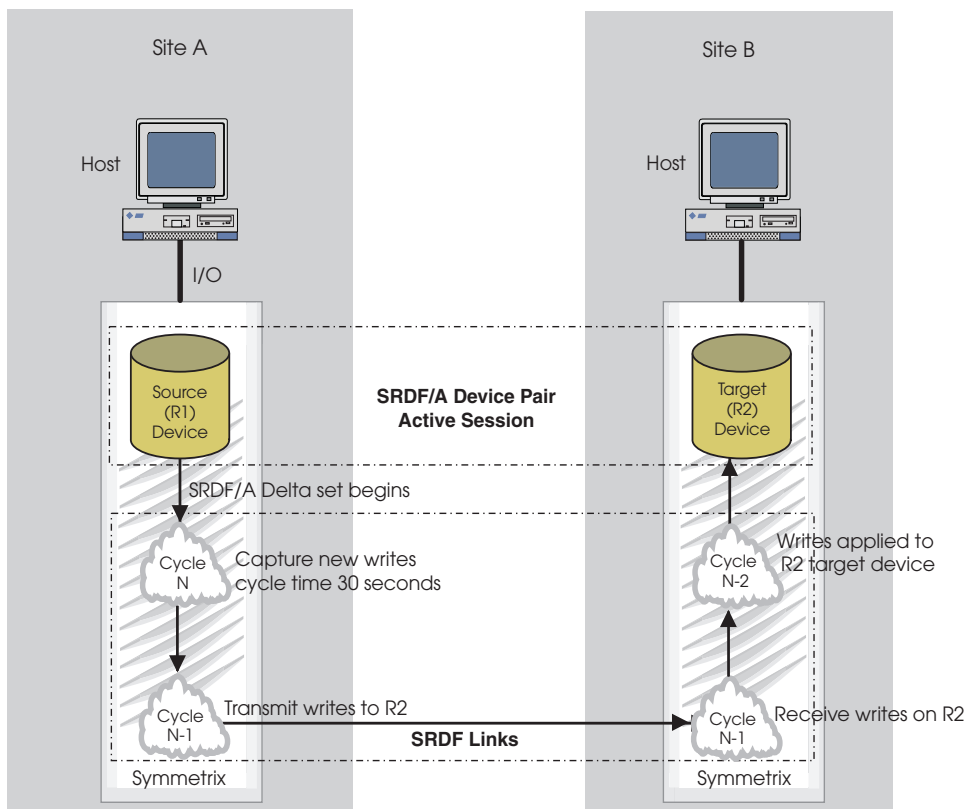


Figure 3-5 SRDF/Asynchronous Mode

Dependent write consistency is achieved through the processing of the ordered SRDF/A delta sets between the source (R1) and the target (R2). Refer to Figure 3-5 on page 3-13, which depicts the SRDF/A cycles. Dependent write consistency ensures that all writes to the R2 are processed in sequential numbered sets to maintain a consistent copy of data between the R1 and the R2.

When the first SRDF/A cycle (N) is active, it collects any new writes on the R1, overwriting any duplicate tracks intended for data transfer over the link. The cycle is active for a predetermined amount of time, which can be configured on the Symmetrix array. The default time is 30 seconds. After the set time has been reached, the delta set data moves into the next cycle position ($N-1$) and begins transferring the delta set over the link to the R2. A new cycle N then begins collecting new writes again for the next delta set transfer.

In cycle $N-1$, the delta set is temporarily collected on the R2 side for destaging. When the $N-1$ cycle has finished transferring data into the R2 and the minimum cycle time has elapsed, the delta set data moves into the next cycle position ($N-2$) and begins destaging the data to the R2 storage devices. The delta set data is considered committed to the R2 in cycle $N-2$ as it is applied to disk.

One delta set is dependent upon the other for achieving write consistency. No cycle can begin until the prior one has completed. All data is transferred at the block level.

Note: The cycle is elongated if the write transfer or destaging exceeds the set cycle time.

When all delta set $N-2$ data is applied to the R2 target device, the R1 and R2 are considered to be in the *consistent pair state*, both containing a consistent image of data. The user can verify if the data in a current SRDF/A session has been applied to the R2 by using the `symrdf checkpoint` command. Refer to *Confirm R2 Data Copy* on page 2-46.

Consistency protection can be enabled and disabled for SRDF/A-capable devices. Refer to *Enable Consistency Protection* on page 2-45 to enable consistency for devices managed by device group or device file.

For instructions on how to enable consistency for composite group operations, refer to *RDF Consistency Group Operations* on page 3-55.

SRDF/A Session Status

When asynchronous mode is set for a group of devices, the SRDF/A-capable devices in the group are considered part of the SRDF/A session. The session status is displayed as active or inactive:

- ◆ Inactive — This status indicates that 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).
- ◆ Active — This status indicates that the SRDF/A mode is activated and that SRDF/A session data is currently being transmitted in operational cycles to the R2.

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.

Use the `symdg show` command to display SRDF/A session status information. SRDF/A session status information appears as follows:

```
R D F A I n f o r m a t i o n :
{
  S e s s i o n   N u m b e r                : 10
  C y c l e   N u m b e r                   : 25
  N u m b e r   o f   D e v i c e s   i n   t h e   S e s s i o n : 15
  S e s s i o n   S t a t u s                : A c t i v e

  S e s s i o n   C o n s i s t e n c y   S t a t e           : N / A
  M i n i m u m   C y c l e   T i m e                   : 00:00:30
  A v e r a g e   C y c l e   T i m e                   : 00:00:30
  D u r a t i o n   o f   L a s t   C y c l e             : 00:00:30
  S e s s i o n   P r i o r i t y                     : 33

  T r a c k s   n o t   C o m m i t t e d   t o   t h e   R 2   S i d e : 1234
  T i m e   t h a t   R 2   i s   b e h i n d   R 1         : 00:00:45
  R 1   S i d e   P e r c e n t   C a c h e   i n   U s e   : 0
  R 2   S i d e   P e r c e n t   C a c h e   i n   U s e   : 0
}
```

Note: Display formats have also been updated for the `symdg show` command to include SRDF/A status information. Refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide* for more information.

Listing SRDF/A Device Information

You can use the `symrdf list -rdfa` command to list SRDF/A-capable devices. When the `-rdfa` parameter is specified, only the SRDF/A devices are displayed. 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.

Note: Beginning with Enginuity 5671, all devices are SRDF/A-capable and the `symrdf list -rdfa` command will display all devices.

The following is an example of the `symrdf list -rdfa` command output.

symrdf list -sid 493 -rdfa

Symmetrix ID: 000187900493

Local Device View

		STATUS		MODES		RDF STATES				
Sym	RDF					R1 Inv	R2 Inv			
Dev	RDev	Typ:G	SA	RA	LNK	MDA	Tracks	Tracks	Dev	RDev Pair
0000	0000	R1:1	??	RW	RW	A..	0	0	RW	WD Consistent
0001	0001	R1:1	??	RW	RW	A..	0	0	RW	WD Consistent
0002	0002	R1:1	??	RW	RW	A..	0	0	RW	WD Consistent
0003	0003	R1:1	??	RW	RW	A..	0	0	RW	WD Consistent
0004	0004	R1:1	??	RW	RW	A..	0	0	RW	WD Consistent

The RDF query command should be used to display SRDF/A group information, which includes the asynchronous mode of operation and Consistent state of devices. For example:

symrdf -g GroupA query -rdfa

```

Device Group (DG) Name      : GroupA
DG's Type                   : RDF1
DG's Symmetrix ID           : 000000006163
RDFA Session Number         : 10
RDFA Cycle Number           : 100
RDFA Session Status          : Active
RDFA Minimum Cycle Time     : 00:00:30
RDFA Avg Cycle Time          : 00:00:30
Duration of Last Cycle       : 00:00:30
RDFA Session Priority        : 33
Tracks not Committed to the R2 Side : 1234
Time that the R2 is behind the R1 : 00:00:40
RDFA R1 Side Percent Cache In Use : 0
RDFA R2 Side Percent Cache In Use : 0

```

Source (R1) View						Target (R2) View				MODES	
		ST			LI			ST			
Standard		A			N			A			
Logical		T	R1 Inv	R2 Inv	K			T	R1 Inv	R2 Inv	
Device	Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	MDAC	RDF Pair
DEV001	00F2	RW	0	0	RW	00E6	WD	0	0	A...	Consistent
DEV002	00F3	RW	0	0	RW	00E7	WD	0	0	A...	Consistent
DEV003	00F4	RW	0	0	RW	00E8	WD	0	0	A...	Consistent
DEV004	00F5	RW	0	0	RW	00E9	WD	0	0	A...	Consistent
DEV005	00F6	RW	0	0	RW	00EA	WD	0	0	A...	Consistent
DEV006	00F7	RW	0	0	RW	00EB	WD	0	0	A...	Consistent
DEV007	00F8	RW	0	0	RW	00EC	WD	0	0	A...	Consistent
Total											
Tracks			0	0		0			0	0	
MB(s)			0.0	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, - = N/A

```

Using the Immediate Option

The `-immediate` option applies to SRDF/A-capable devices when used with the `failover`, `split`, or `suspend` commands. When applied with these commands, the SRDF/A session will be immediately dropped and the devices will be made Not Ready on the link. The devices will remain in asynchronous mode and any pending tracks will be converted to invalid tracks.

Using this option will most likely 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 `symrdf` commands. Refer to Chapter 2 for *Singular SRDF Control Operations* and the *Invalid Tracks* on page 2-30.

By default, issuing a `failover`, `split`, or `suspend` command without the `-immediate` option will cause the SRDF/A session to be dropped and the devices to be made Not Ready on the link at the end of the current cycle. Execution time of the command may be elongated, but yields no remote invalid tracks on the R2 side.

Note: The `symrdf query -rdfa` option displays the number of tracks not committed to the R2 side as well as any invalid tracks.

Note: If consistency is enabled on SRDF/A-capable devices within the group, then the `-force` option must be applied for the `failover`, `split`, and `suspend` commands.

Using BCVs to Preserve R2 SRDF/A Data Copy

Although not required for SRDF/A mode, it is recommended that you use TimeFinder BCVs at the remote site to mirror R2 devices and preserve a consistent image of data before resynchronization operations. R2 device BCVs can be consistently split off of the R2 without having to drop the RDF links and without disruption 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. Controlling the R2 BCVs from the R1-side host requires using the `symmir` command with the `-rdf` option.

For example, to consistently split off the R2 SRDF/A-capable device BCVs in group `prod` from the R1 host, enter:

```
symmir -g prod split -rdf -consistent
```

Note: For more information on the `symmir -consistent split` command, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

Mode Transition to Synchronous

Solutions Enabler version 6.0 supports a consistent mode transition from asynchronous to synchronous for devices managed by device group or device file. A consistent mode transition preserves database consistency for data on the R2 side. This functionality requires Enginuity 5671.

For example, to switch modes from asynchronous to synchronous and maintain R2 data consistency in group `prod`, enter:

```
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`, enter:

```
symrdf -f devfile1 -consistent set mode sync
```

Note: Completion of a consistent mode transition requires two SRDF/A cycle switches.

Dynamic SRDF Group Operations

You can add, modify, and delete dynamic RDF groups (RA groups) in a switched fabric SRDF environment. Previously, *static* RDF groups could only be defined at the time of unit configuration. RDF groups can now be created on demand while the Symmetrix array is in operation.

RDF groups provide communication paths linking two separate Symmetrix arrays. These communications paths are used to synchronize data between the R1 and R2 device pairs associated with the group. At least one physical connection must exist between the two Symmetrix arrays within the fabric topology. Refer to *RDF Groups in a Switched SRDF Link* on page 3-8 for an example.

Physical point-to-point fibre connections are not currently supported, even if the RDF connections are configured as "switched". For information on RDF groups using point-to-point fibre connections, refer to *RDF Groups in a Point-to-Point SRDF Link* on page 3-7.

Dynamic SRDF group capability provides flexibility within your SRDF environment to change multiple remote mirroring connections for dynamic devices.

Note: Dynamic group capability is only supported in Enginuity 5669 and higher. Dynamic RDF devices cannot currently be used with Parity RAID.

Adding Dynamic Groups

Using the SYMCLI `symrdf addgrp` command you can create a dynamic RDF group that represents an additional RDF link between two Symmetrix arrays.

Since Enginuity 5669, up to 64 RA groups are allowed, which can execute RDF control actions in parallel. Adding a dynamic RDF group creates an empty group. Dynamic groups must be added one at a time. Adding multiple dynamic groups can be executed in a script, but must be done one group at a time. Once the group is created, you can then add dynamic SRDF pairs to it. A group label must be specified when adding a dynamic group. Only one dynamic group operation (`addgrp`, `modifygrp`, `removegrp`) can be executed at a time before another can be attempted.

Note: Before you can add a dynamic RDF group, the `dynamic_rdf` parameter must be set in your Symmetrix configuration. Refer to the *Symmetrix Configuration Change CLI Product Guide* for information on how to set Symmetrix metrics.

The following example adds a new dynamic RDF group (`dynggrp4`), which represents the RDF link between two Symmetrix arrays (6180 and 6240). It adds dynamic RDF group 4 on the local Symmetrix array (6180), and RDF group 4 on the remote Symmetrix array (6240). The specified group label (`dynggrp4`) can later be used to modify or delete the group. Directors are specified for both the local (12a) and remote (13a) Symmetrix arrays.

```
symrdf addgrp -label dynggrp4 -rdfig 4 -sid 80 -dir 12a \
  -remote_rdfg 4 -remote_sid 40 -remote_dir 13a
```

When creating dynamic SRDF groups between two Symmetrix arrays, it is important to understand the network topology when choosing director endpoints. If using fibre protocol, the director endpoints chosen must be able to see each other through the Fibre Channel fabric in order to create the dynamic RDF link. Ensure that the physical connections between the local RA and remote RA are valid and operational. Static and dynamic groups may co-exist on the same RA directors.

Note: It is recommended that you configure no more than 6 groups per switched fibre RA as performance could be degraded.

After the group has been created, dynamic SRDF pairs can be added to it. The following example adds the dynamic SRDF pairs listed in the device file (`dynpairsfile`) to the new dynamic SRDF group 4.

```
symrdf createpair -file dynpairsfile -sid 80 -rdfig 4
  -type rdf1 -invalidate r2
```

For more information on adding dynamic RDF pairs, refer to *Dynamic SRDF Pair Operations* on page 2-59.

You can use the `symcfg list -ra all -switched` command to display all RDF groups on the local Symmetrix array and its remotely connected Symmetrix arrays. RDF groups are listed as static or dynamic under the group type.

Modifying Dynamic Groups

Using the `symrdf modifygrp` command, you can modify an existing Dynamic RDF group. The modify command can be used to add or remove supporting directors to a dynamic RDF group. Reassigning directors for RDF dynamic groups requires that you understand the network fabric topology when choosing director endpoints. The modify command cannot be used to modify existing static groups. You must specify the group label or group number.

The following example modifies a dynamic group (`dynggrp4`) to remove a supporting director 13a assigned from the group on the local Symmetrix array 6180.

```
symrdf modifygrp -label dynggrp4 -sid 80 \  
-remove -dir 13a
```

The following example modifies a dynamic group (`dynggrp4`) to add (assign) a supporting director 12a to the group on the local Symmetrix 6180.

```
symrdf modifygrp -label dynggrp4 -sid 80 \  
-add -dir 12a
```

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



CAUTION

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

Removing Dynamic Groups

A dynamic SRDF group must first be emptied of its assigned devices using the `symrdf deletepair` command before it can be removed. At least one physical connection between Symmetrix arrays must exist. Deleting the dynamic group removes all local and remote director support.

The following example deletes from the group the SRDF dynamic pairs defined in a device file, and then removes the local and remote dynamic SRDF groups created and modified in the previous examples.

```
symrdf deletepair -file dynpairsfile -sid 80 -rdfg 4  
  
symrdf removegrp -sid 80 -label dynggrp4
```



CAUTION

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

Concurrent RDF Operations

In an SRDF configuration, a single source (R1) device can concurrently be remotely mirrored to two target (R2) devices. This feature, available with Enginuity 5567-based Symmetrix arrays and higher, is known as a concurrent RDF configuration and is supported with ESCON and fibre interfaces. This allows you to have two identical remote copies available at any point in time. It is valuable for duplicate restarts or disaster recovery, or for increased flexibility in data mobility and migrating applications.

Concurrent RDF technology can use two different RA adapters (RAs, RAFs, or RFs) in the interface link to achieve the connection between the R1 device and its two concurrent R2 mirrors. Each of the two concurrent mirrors must belong to a different RDF (RA) group. As illustrated in Figure 3-6, RDF Group 1 is one link to remote Site B and RDF Group 2 is the second link to remote Site C.

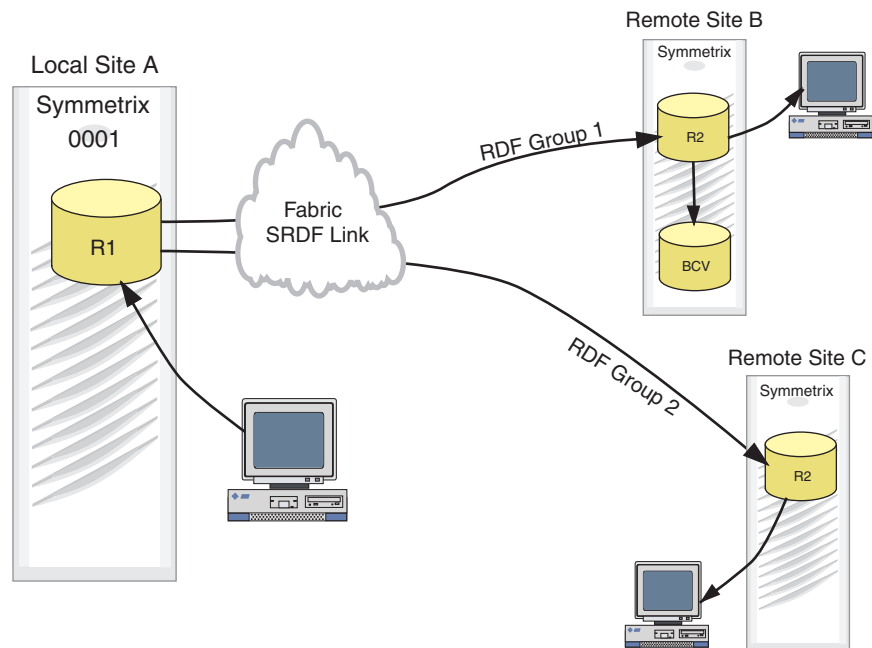


Figure 3-6 Concurrent RDF

Supported Concurrent RDF Modes

Each of the two remote mirrors of a concurrent RDF configuration can operate independently (but concurrently) in any of the following SRDF modes:

- ◆ Synchronous
- ◆ Semi-synchronous
- ◆ Adaptive Copy

The modes for these two mirrors can be the same or different, except you cannot have one mirror in synchronous and the other in semi-synchronous mode.

Beginning with Enginuity 5671, SRDF/Asynchronous mode is supported for SRDF/A-capable devices in a concurrent RDF configuration. Only one RDF group in a concurrent configuration can be operating in asynchronous mode at one time. The other RDF group must be in a mode other than asynchronous (i.e., synchronous, semi-synchronous, or adaptive copy).

With the exception of *failback*, *restore*, *swap*, and *R1 update* operations, all composite SRDF control operations can be performed on a concurrent RDF configuration. In these three exception operations, they cannot be performed concurrently because data cannot be simultaneously copied from two R2 devices to the same R1 device.

Note: The following cannot be (or are not supported as) concurrent RDF devices:

- Dynamic RDF devices prior to Enginuity 5670
- RDF consistency group operations or part of a composite group

Device Groups and RDF Groups

With concurrent RDF, you can build a device group containing standard devices that belong only to the two RDF groups representing the concurrent remote mirrors. Your device group can also include BCV devices and RDF standard devices that are not concurrent RDF devices. However, within the context of the device group, you can remotely associate a BCV with only one of the concurrent R2 mirrors (as shown in Figure 3-6), not both.

RDFG Option

When controlling or setting concurrent RDF devices, the `-rdfg n` option is required to specify which RDF (RA) group number (*n*) or remote mirror of the R1 device is to be controlled. If the operation is to be performed on both concurrent remote mirrors, then `-rdfg ALL` should be used.

Viewing Concurrent RDF Devices

Using the `-concurrent` option with `symrdf list`, you can view all the Symmetrix devices to see which were configured as concurrent RDF devices:

```
symrdf list [-sid SymmID] -concurrent
```

Using the `-rdfg ALL` option with `symrdf query`, you can view the RDF states and modes of both remote mirrors of a concurrent RDF device:

```
symrdf -g DgName query -rdfg ALL
```

Establishing Concurrent RDF Devices

To create a device group for the concurrent RDF devices and initially synchronize (establish) the devices across the concurrent RDF links, follow these steps:

1. Create a concurrent RDF device group:

```
symdmg create ConcGrp -type RDF1
```

2. Add all concurrent RDF devices to the device group:

```
symld add dev 0001 -g ConcGrp -sid 0001
symld add dev 0021 -g ConcGrp
symld add dev 002A -g ConcGrp
.
.
.
```

3. Establish concurrent SRDF pairs that belong to the device group (first one remote mirror and then the other):

```
symrdf -g ConcGrp establish -rdfg 1
symrdf -g ConcGrp establish -rdfg 2
```

Or, you can use the `-rdfg ALL` option to simultaneously establish both remote mirrors of each SRDF pair in one command:

```
symrdf -g concGrp -full establish -rdfg ALL
```

Splitting Concurrent RDF Devices

You split the concurrent SRDF pair either simultaneously or sequentially. To split the links simultaneously, enter:

```
symrdf -g concGrp split -rdfg ALL
```

To split the two remote mirrors one at a time, enter:

```
symrdf -g concGrp split -rdfg 1
symrdf -g concGrp split -rdfg 2
```

Restoring Concurrent RDF Devices

If you need to restore data from the target R2 devices to the source R1 devices, only one of the concurrent RDF R2 mirrors must be selected as the mirror from which to restore. (This rule applies to failback and R1 update actions as well.) In the following example, both remote mirrors are split and the R1 device is being restored from the R2 device in RDF Group 1:

```
symrdf -g concGrp restore -rdfg 1
```

As shown in Figure 3-7, after the restore operation, the R1 device is synchronized with the R2 mirror belonging to RDF Group 1, and the R2 device belonging to RDF Group 2 is still in a split state. The devices belonging to RDF Group 2 can now be re-established to be in a synchronized concurrent RDF state.

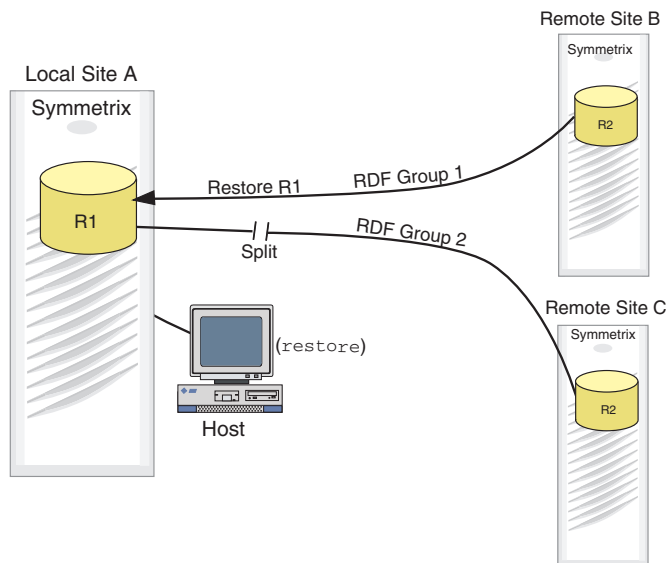


Figure 3-7 Restoring R1 in a Concurrent RDF

The Remote Option for Restore, Update, Failback

The remote data copy option (`-remote`) applies to the failback, restore, and R1 update operation typically when you want to restore data to both the R1 device and the other R2 mirror. As shown in Figure 3-8, once the R1 is restored, then the other R2 mirror is restored.

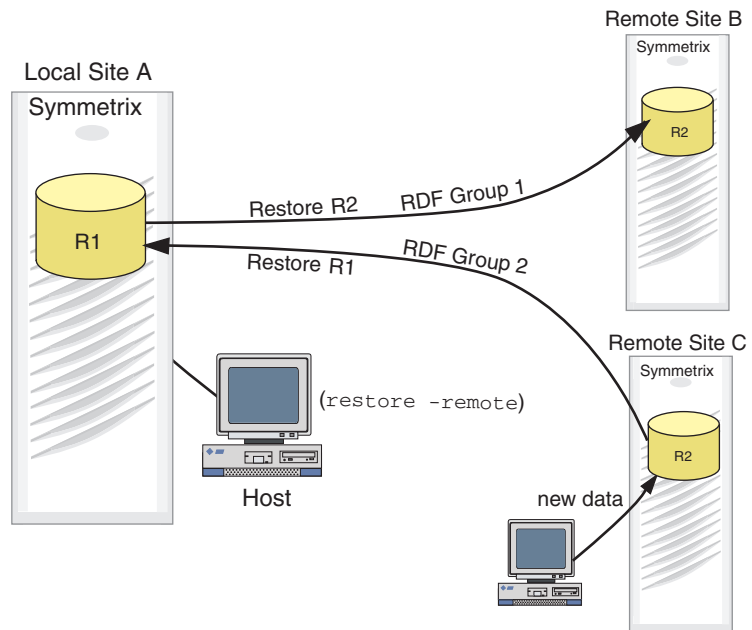


Figure 3-8 Restoring R1 and the Other R2 in a Concurrent RDF

The following example illustrates the remote data copy restore operation:

```
symrdf -g ConcGrp restore -rdfg 2 -remote
```

In this example, the data propagates all the way from the R2 mirror of RDF Group 2 to the R1 device, and then to the other R2 mirror, which synchronizes all concurrent RDF mirrors.

Note: Since Enginuity 5670, the `-remote` option can be applied with the `createpair` command for a restore operation to dynamically create a concurrent RDF pair by adding a second remote mirror. Refer to *Creating Dynamic Concurrent SRDF Pairs* on page 2-63.

TimeFinder Consistent Splits Across RDF

TimeFinder® consistent split allows you to split off a consistent, restartable copy of a database management system within seconds with no interruption to online service. A concurrent split helps to avoid inconsistencies and restart problems that can occur when splitting a database-related BCV without first quiescing the database. This functionality requires a TimeFinder/CG license. Consistent split can be implemented using either PowerPath-connected devices or the Enginuity Consistency Assist feature. For information about consistent split using the Enginuity Consistency Assist feature, refer to *Enginuity Consistency Assist (ECA)* on page 3-29.

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.

Consistent Split on Both RDF Sides Using PowerPath

In an RDF environment as shown in Figure 3-9, you can perform an Enterprise TimeFinder consistent split, using PowerPath devices to split the BCVs in both the local Symmetrix array and the remote Symmetrix array.

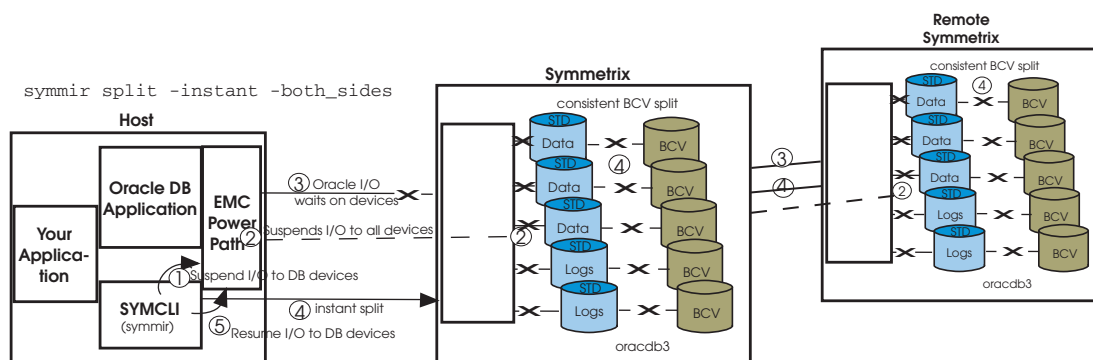


Figure 3-9 Consistent Split on Both Sides

Consistent split actions on both sides of the RDF links are only implemented with an instant (`-instant`) split where you must also specify a database or PowerPath device(s).

To target a database target (both sides), use the following syntax:

```
symmir -g DgName split -instant -both_sides
-rdb -dbtype DbType [-db DbName]
[-preaction Script][-postaction Script]
```

To target the PowerPath standard devices of the group, or just specific PowerPath device names (both sides), use the following syntax:

```
symmir -g DgName split -instant -both_sides
-ppath STDDEVS|<PowerPathPdevName...>
[-preaction Script][-postaction Script]
```

For more information about the `symmir` command, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

Enginuity Consistency Assist (ECA)

Since Enginuity 5568, you can use the Enginuity Consistency Assist (ECA) feature to perform consistent split operations across multiple, heterogeneous hosts without the use of PowerPath support.

TimeFinder consistent split operations are accomplished using the `-consistent` option with the `symmir` command. For more information about the `symmir` command, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

The `-consistent` option can also be used with the `symreplicate` command to run a copy cycle, which freezes I/O to all devices in a device or composite group for both single-hop and multi-hop configurations.

To consistently split BCV pairs using ECA you must have either a control host with no database or a database host with a dedicated channel. Refer to Figure 3-10 on page 3-30 for a depiction of how a control host can perform ECA consistent splits for three database hosts that access devices on a Symmetrix array.

Symmetrix device groups must be created on the controlling host for the target database to be consistently split. Device or composite 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.

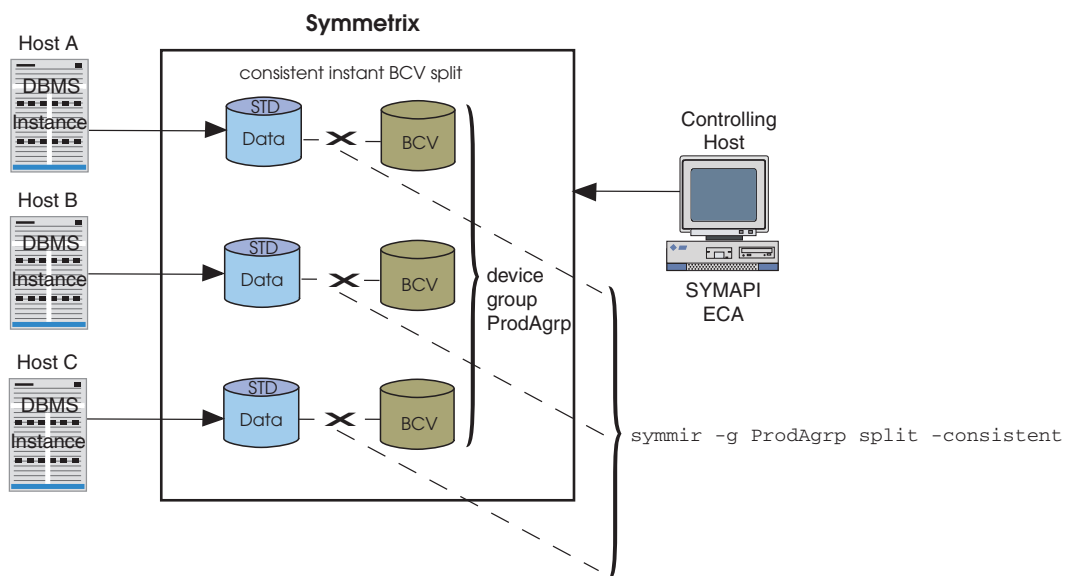


Figure 3-10 ECA Consistent Split

SRDF/Automated Replication Operations

The `symreplicate` command invokes the SRDF Automated Replication (SRDF/AR) facility. The command performs automated consistent replication of data from standard devices via RDF1 BCV devices over SRDF links. This functionality requires SRDF, SRDF/AR (since 5670.55), and TimeFinder/Mirror licenses. For consistent split operations, a TimeFinder/CG license is also required. By default, the replication process is performed in the background. The `symreplicate` command supports both single-hop and multi-hop SRDF configurations. You can start, stop, or restart a replicate session without degradation of the data copy. During a replication session, you can have access to an independent copy of the replicating data by setting up a concurrent BCV.

Note: For a full description of the `symreplicate` command syntax, refer to the *EMC Solutions Enabler Symmetrix Command Reference Manual*.

Note: SRDF/AR does not support SRDF/Asynchronous-capable devices.

Locked Devices

Device external locks in the Symmetrix array are held during the entire replicate session, which is necessary to block other applications from altering device states while this session executes. Under certain circumstances, a replicate session may exit with the devices left in a locked state. Device locks can be recovered, released or acquired to be persistent.

Recovering locks

If a replicate session terminates when an RDF link goes down unexpectedly, the replicate session cannot restart after the RDF link is brought back up, because of the locked devices. You can use the `-recover` option with the `symreplicate start` or `restart` command to recover the device locks and restart the session.

Note: As long as the exact same devices are still locked under the lock holder ID of the previous replicate session, then the device locks can be recovered.

Releasing Locks

Beginning with Solutions Enabler 6.0, you can optionally release the device external locks held in the Symmetrix 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.

For example, to release devices locks on a terminated session for device group `prod` on Symmetrix 35002, enter:

```
symreplicate -g prod release -sid 35002
```

When the above command is executed, any device external locks associated with devices in device group `prod` that were locked via the previous SRDF/AR session that are still held will be released.

The following restrictions apply to releasing locks:

- ◆ The SRDF/AR session for the targeted devices must not be active.
- ◆ Devices must have been locked by the previous session and the lock holder ID must match the previous session's ID.
- ◆ The number of devices to be unlocked must be less than or equal to the total number of devices in the previous SRDF/AR session.

The force (`-force`) option will be 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 the lock holder ID for some of the devices in the targeted SRDF/AR session do not match the lock holder ID of that session and the user wishes to release the devices locked with the session's original lock holder ID.

Acquiring Persistent Locks

If running the base daemon (SYMAPI daemon), device locks will automatically be released in the event of a system crash or component failure. Optionally, the device locks may be acquired using the persistent attribute by setting the `SYMCLI_REPLICATE_PERSISTENT_LOCKS` parameter to `TRUE` in the `symreplicate` options file. Refer to *Symreplicate File Parameters* on page 3-49 for additional information.

Single-Hop Data Copies

As shown in Figure 3-11, for a single-hop configuration in a complete copy cycle, *symreplicate* copies data:

- ◆ From the standard device to the BCV of the local Symmetrix array (1a path).
- ◆ From the BCV device of the local Symmetrix array to the standard device of the remote Symmetrix array (1b path).
- ◆ From the remote standard device to its BRBCV device (1c path).

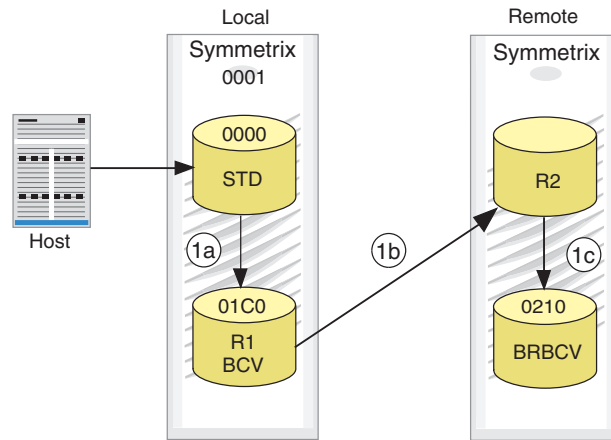


Figure 3-11 Automated Data Copy Path in Single-Hop SRDF Systems

To choose a single-hop replicate session, you must set the replication type parameter in the replicate options file (refer to *Symreplicate File Parameters* on page 3-49) as follows:

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
```

Along the way, the replicate session will incrementally establish RDF and BCV pairs, and then differentially split BCV pairs so that data transfers are reduced.

Setup for a Single-Hop Data Replication

The single-hop data replication copies data from the local Symmetrix array to the remote Symmetrix array. To set up a single-hop data replication session, select any number of standard devices of the same type (R1, R2, or non-RDF), and create a device group or composite group of the same type. Add the devices, and associate an equal number of R1-BCV devices of matching sizes. Finally, associate an equal number of BRBCV devices (remote BCVs), also of matching sizes.

The following command sequence illustrates this setup:

```
symdg create newdg
symld add dev 0000 -g newdg -sid 35002
symld add dev 0001 -g newdg
.
.
.
symbcv associate dev 01C0 -g newdg
symbcv associate dev 01C1 -g newdg
.
.
.
symbcv associate dev 0210 -g newdg -bcv -rdf
symbcv associate dev 0211 -g newdg -bcv -rdf
.
.
.
```

Note: Since Solutions Enabler version 5.4, the `symreplicate` command supports the use of composite groups (`-cg`) to implement single-hop or multi-hop configurations for devices that span multiple Symmetrix arrays.

Before starting a replicate session, the following conditions must be met:

- ◆ Both sets of BCV pairs must have a pairing relationship.
- ◆ The local BCV pairs must be established, the RDF pairs must be in the Suspended pair state, and the remote BCVs (BRBCVs) must be split pair state. Ensure there are no writes allowed to the BRBCV by any directly attached host at the remote site.

Pair State Auto Setup for SRDF/AR

The pair state setup for SRDF/AR can be achieved automatically by using either the `symreplicate setup` command or the `-setup` option with the `symreplicate start` command.

The auto-replication setup action sets up the required pair states for devices and executes one copy (auto-replication) cycle. By setting up the device states ahead of time, replication processing time is saved. The `symreplicate` options file defines the hop type (single or multi) and any copy cycle parameters to be used for the `setup` and `start` commands. The `setup` command executes only one cycle of the replication session, regardless of the number of cycles defined in the options file and then exits.

If you prefer, these conditions can be established by manually reproducing the single-hop replication cycle through a sequence of SRDF and TimeFinder CLI commands. For more information on how to manually set up the single-hop replication environment, refer to *Manual Setup for Single Hop* on page 3-36. Or, for more information on how to manually set up the multi-hop replication, refer to *Manual Setup for Multi-Hop* on page 3-39.

Note that the setup operation only corrects pair states of devices in the group. If a BCV in the group is paired with a standard outside of the group, setup will not correct it.

The following command shows how to execute the `symreplicate setup` command on a device group (`DevGrp1`) using an options file (`OpFile`):

```
symreplicate -g DevGrp1 setup -options Opfile
```

Note: The setup command may take some time to run to completion as it finally exits when devices are in the required pair state for running the replication session.

Note: For more information on the available parameters that can be defined in the replicate options file, refer to *Symreplicate File Parameters* on page 3-49.

When executing the `symreplicate start` command with the `-setup` option, the first cycle puts the devices in the required pair state. The following command line shows how to execute the `symreplicate start` command with the `-setup` option:

```
symreplicate -g DevGrp1 start -options Opfile -setup
```

The default setup operation (either using the `setup` action or the `-setup` option) provides no I/O optimization or engages 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.

Exact Initial Pairing

Or, using the `-exact` option, you can start the replication session with the STD-BCV pair relationships in the exact order that they were associated/added to the device group or composite group.

Optimizing I/O with Pair Assignment

Or, you can optimize the disk I/O on standard/BCV pairs in the device or composite group, using the `-optimize` option when you use the `-setup` option or the `setup` argument. This will cause the setup action to split all pairs and perform an optimized STD-BCV pairing within the specified group.

For device groups using this optimize option, the device pair selection attempts to pair devices in the group that are not on the same disk adapter to distribute the I/O. For example:

```
symreplicate setup -g DgName -optimize
```

For composite groups, the same optimize pairing behavior can be targeted to a Symmetrix RA group. For pair assignment in RA groups that provides remote I/O optimization (distribution by using different remote disk adapters), you use the `-optimize_rag` option with either the `-setup` option or the `setup` argument. For example:

```
symreplicate setup -cg CgName -optimize_rag
```

Note: Single-hop replication does a full optimization on all RA groups.

Consistent Split Option

Using the `-consistent` option with the `start` action, you can consistently split all of the BCV pairs on the local Symmetrix array for a typical SRDF configuration, or on the Hop 1 remote Symmetrix array for a multi-hop configuration. This also requires a TimeFinder/CG license.

Manual Setup for Single Hop

If you prefer, these conditions can be established by manually reproducing the single-hop replication cycle through a sequence of SRDF and TimeFinder CLI commands. The following are the manual single-hop replication steps:

1. After waiting for any ongoing establish to complete, split the BCV pairs.

```
symmir split -g newdg
```

2. Establish the RDF pairs.

```
symrdf establish -g newdg -bcv
```

3. After waiting for any ongoing establish to complete, suspend the RDF pairs.

```
symrdf suspend -g newdg -bcv
```

4. Establish the BCV pairs.

```
symmir establish -g newdg -exact
```

5. Establish the remote BRBCV pairs.

```
symmir establish -g newdg -bcv -rdf -exact
```

6. After waiting for any ongoing establish to complete, 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 (i.e., `establish -full` for BCV pairs without relationships).

The `-preaction` and `-postaction` options can be used to specify scripts for `symreplicate` to run before and after step 1 (splitting the BCVs).

Multi-Hop Data Copies

As shown in Figure 3-12 on page 3-38, for a multi-hop configuration in a complete copy cycle, `symreplicate` copies data:

- ◆ From the local standard device to the standard of the remote Hop 1 Symmetrix (2a path).
- ◆ From the Hop 1 standard device to its BCV (RBCV) (2b path).
- ◆ From the Hop 1 RBCV device to the standard device of Hop 2 Symmetrix (2c path).
- ◆ From the Hop 2 standard device to its BCV (RRBCV) (2d path)¹.

To choose a multi-hop replicate session, you must set the replication type parameter in the replicate options file (refer to *Symreplicate File Parameters* on page 3-49) as follows:

```
SYMCLI_REPLICATE_HOP_TYPE=MULTI
```

Note: If you do not want the final Hop 2 BCV updated, you can set `SYMCLI_REPLICATE_USE_FINAL_BCV=FALSE` in the replicate options file.

1. Applies only when you have a final BCV in this Hop 2 Symmetrix path and you have not disabled it.

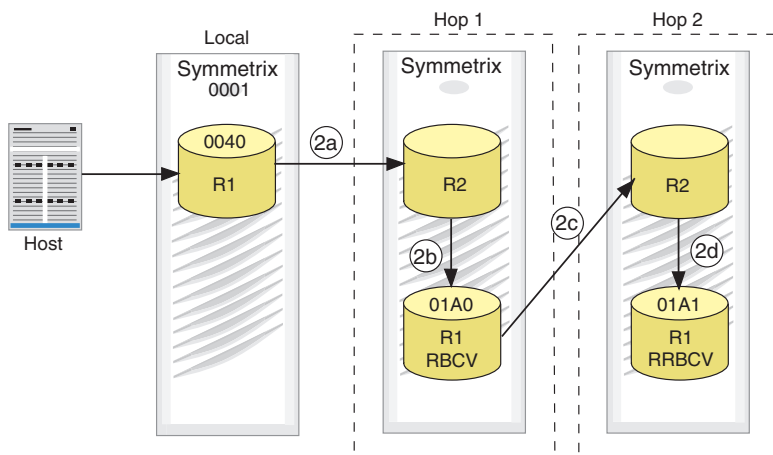


Figure 3-12 Automated Data Copy Path in Multi-Hop SRDF Systems

Setup for a Multi-Hop Data Replication

Multi-hop data replication copies data from the local Symmetrix array to the Hop 1 remote Symmetrix array, and then to the Hop 2 Symmetrix array. To set up a multi-hop data replication session, create an R1 device group (-g) or composite group (-cg), and add any number of R1 devices. Remotely associate an equal number of matching sized R1-BCVs or Hop 1 RBCV devices.

The following command sequence illustrates this setup:

```

symdg create newdg2 -type RDF1
symld add dev 0040 -g newdg2 -sid 0001
.
.
symbcv associate dev 01A0 -g newdg2 -rdf
symbcv associate dev 01A1 -g newdg2 -rrdf

```

Before starting a replicate session without a setup operation, the local RDF pairs must be synchronized, the BCV pairs must be established, and the remote RDF pairs must be suspended. If the final BCVs in the second-hop Symmetrix array are used, the BCVs must be in the split state.

Optionally, the device pair state can be configured automatically by using the `symreplicate setup` command or the `-setup` option with the `symreplicate start` command. Refer to *Pair State Auto Setup for SRDF/AR* on page 3-34 for information on automating setup conditions.

Manual Setup for Multi-Hop

Manual setups for these conditions can be established by manually reproducing the multi-hop replication cycle through a sequence of TimeFinder `symmir` CLI commands. The following are the manual multi-hop replication steps for the configuration in Figure 3-12:

1. After waiting for any ongoing establish to complete, split the BCV pairs (point 2b).

```
symmir split -g newdg2 -rdf -remote
```

Establish the remote RDF pairs (first hop BCV with R2 second hop at point 2c). (This step was accomplished in this last command by the use of the `-remote` option.)

2. After waiting for the RDF establish to complete, suspend the remote RDF pairs (2c), and establish the BCV pairs (2b).

```
symmir establish -g newdg2 -rdf -exact
```

3. Establish the BCV pairs in the second Symmetrix hop (2d) by using either a device file or the `-rrbcv` command option.

```
symmir establish -f 2nd_hop_devs.txt -sid SymmID
or
symmir establish -g newdg2 -rrbcv
```

Note: To use the `-rrbcv` option, the RDF BCV devices must have been previously associated with the group, using `symbcv -rrdf`.

4. After waiting for any ongoing establish to complete, split the 2nd hop BCV pairs.

```
symmir split -f 2nd_hop_devs.txt
or
symmir split -g newdg2 -rrbcv
```

Note: steps 3 and 4 are performed when you want the final hop 2 BCV(s) to be used in the replicate cycle.

Note: You may have to include additional command options in some of the above steps (such as `establish -full` for BCV pairs without relationships).

The `-preaction` and `-postaction` options can be used to specify scripts for `symreplicate` to run before and after step 1 (splitting the BCVs).

Concurrent BCVs With SRDF/AR

If you require an independent copy of your data during a replication cycle, you can set up concurrent BCVs. One BCV copy is included in with the SRDF/AR device group and the other BCV copy is not. The BCV that is not part of the replication cycle receives the identical data as the BCV that is part of the SRDF/AR devices. The non-SRDF/AR BCV can be accessed by its host during the replication cycle.

Concurrent BCV in a Single-Hop Configuration

Refer to Figure 3-13 for a depiction of how to set up concurrent BCVs in a single-hop configuration. The non-SRDF/AR BCV is paired with the SRDF/AR standard device in a device file or by defining a separate device group on a host different from the one defining the SRDF/AR device group. The non-SRDF/AR BCV is then established with the SRDF/AR standard device so that data is copied to both BCVs.

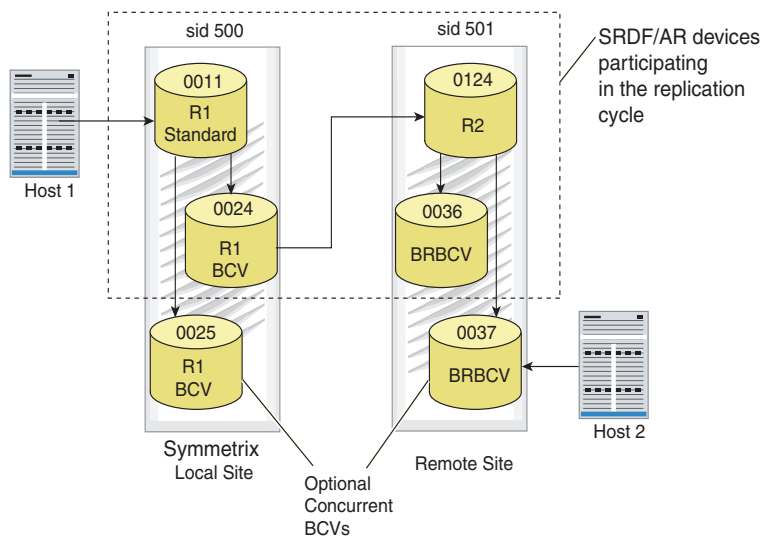


Figure 3-13 Concurrent BCVs in a Single-Hop Configuration

Because participating devices are locked during the replication cycle, a special symmir option (`-skip`) has been implemented to override the lock on the standard device. This allows the BCVs to be concurrently established with the standard device during the SRDF/AR copy cycle.

In Figure 3-13 on page 3-40, BCV 0024 and BRBCV 0036 are concurrent BCVs, as are BCV 0025 and BRBCV 0037. But BCV 0025 and BRBCV 0037 are not included in the replication cycle. A device file can be defined on Host 1 to include the R1 standard device and BCV 0025. These devices can then be established as a BCV pair during the replication cycle, and then split so that the data may be accessed without interfering with the replication cycle.

The following steps explain the previous example for establishing a set of non-SRDF/AR BCVs. All commands are issued from the source-connected host and affect devices on both Symmetrix arrays (sid 500 and sid 501).

1. Create the SRDF/AR device group, add devices, and associate the BCV devices using the following commands:

```
symdg create srdfar
symld -g srdfar add dev 0011 -sid 500
symbcv -g srdfar associate dev 0024
symbcv -g srdfar associate dev 0036 -bcv -rdf
```

2. Create two device files (devfile1 and devfile2) to define the non-SRDF/AR BCV pairs on each Symmetrix array.

```
/*devfile1 */
0011 0025
```

```
/*devfile2 */
0124 0037
```

3. Establish and split the non-SRDF/AR BCV pairs in the device files.

```
symmir -f devfile1 -sid 500 establish -full\
-noprompt
symmir -f devfile1 verify
symmir -f devfile1 -sid 500 split -noprompt

symmir -f devfile2 -sid 501 establish -full\
-noprompt
symmir -f devfile2 verify
symmir -f devfile2 -sid 501 split -noprompt
```

4. Use the following commands to set up the SRDF/AR copy cycle for devices in the device group `srdfar`.

```
symmir -g srdfar establish -full -exact -noprompt
symmir -g srdfar verify
symmir -g srdfar split -noprompt
symmir -g srdfar verify -split
symrdf -g srdfar -bcv establish -noprompt
symrdf -g srdfar -bcv split -noprompt
symmir -g srdfar establish -full -exact -bcv -rdf \
-noprompt
symmir -g srdfar verify
symmir -g srdfar split -bcv -rdf -noprompt
symmir -g srdfar verify -split
symmir -g srdfar establish -noprompt
symmir -g srdfar verify
```

Note: Use the `verify` action to check the device state between the `establish` and `split` actions.

5. Check the status of devices in the device group `srdfar`.
6. Begin running the SRDF/AR copy cycle in the foreground using the `symreplicate` command.

```
symreplicate -g srdfar -foreground start -noprompt
```

7. Split off the non-SRDF/AR BCV pairs using the `-skip` option to bypass the locks on the standard devices.

```
symmir -f devfile1 -sid 500 split -instant \
-noprompt -skip
symmir -f devfile2 -sid 501 split -instant \
-noprompt -skip
```

8. Check the status of non-SRDF/AR devices.

```
symmir -f devfile1 -sid 500 query -multi
symmir -f devfile2 -sid 501 query -multi
```

Concurrent BCV in a Multi-Hop Configuration

Refer to Figure 3-14 for a depiction of how to set up concurrent BCVs in a multi-hop configuration. 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 then used to establish the BCV pairs, split them, and access the BCV devices.

For more information, refer to the *Using SYMCLI to Perform Symmetrix RDF Automated Replication* white paper.

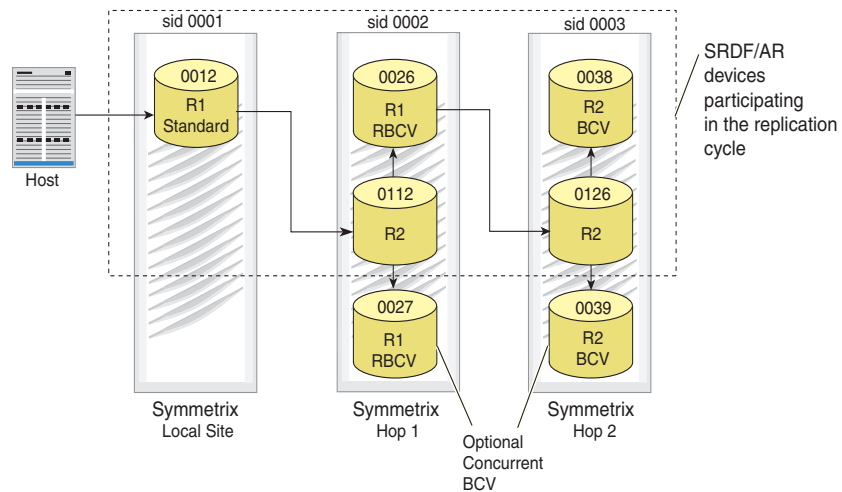


Figure 3-14 Concurrent BCV in a Multi-Hop Configuration

Replication Cycle Patterns

You can manipulate the replication cycle patterns to fit your site's needs by setting the following parameters in the symreplicate options file (refer to *Symreplicate File Parameters* on page 3-49 for options file syntax) as follows:

```
SYMCLI_REPLICATE_CYCLE=CycleTime
```

CycleTime is a timer that indicates the period of time in *minutes* or *hours:minutes(hh:mm)* between when each copy action starts to when it starts again (how often the copy reoccurs). For example, a *CycleTime* of 120 would kick off a new copy every 2 hours.

`SYMCLI_REPLICATE_NUM_CYCLES=NumCycles`

NumCycles indicates the number replication cycles (copies) to perform before `symreplicate` exits. For example, a zero value (the default value) results in continuous cycling until the `symreplicate stop` command is issued.

`SYMCLI_REPLICATE_CYCLE_DELAY=Delay`

Delay indicates the minimum amount of time to wait between the end of one copy cycle and the beginning of the next. For example, a *Delay* of 20 would always force a wait of 20 minutes or more between cycles.

`SYMCLI_REPLICATE_CYCLE_OVERFLOW=OvfMethod`

OvfMethod indicates how to behave when the actual copy time of your data and/or data transfer throughput is so large as to exceed the *CycleTime* value. Here, the initial copy event has overflowed into the period that should be for the next copy cycle. Possible behavior values are:

IMMEDIATE — When overflowed, starts a new cycle immediately after the current copy finishes.

NEXT — When overflowed, waits for the copy to finish, and then starts at the next expiration time (*CycleTime*). (Starts the copies on multiples of the *CycleTime* parameter.)

For example, if a 1-hour copy cycle completed in 1.5 hours, the next cycle could be set to begin immediately or in half an hour (NEXT).

First Time Cycle Parameters

Choosing all the exact cycle time parameters (described in the previous section) may not be possible the first time. A basic replication parameter strategy is to loosely select time constraints, and then tighten the parameters at some point later when you have a sense of data size and SRDF throughput expectations.

The following are two possible parameter setups for an initial replicate session trial:

Table 3-2 Initial Setups for Cycle Timing Parameters

SYMCLI_REPLICATE_CYCLE=60 SYMCLI_REPLICATE_CYCLE_DELAY=0 SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT	Every hour if possible. Else, every 2, or 3 hours based on data throughput and size.
SYMCLI_REPLICATE_CYCLE=0 SYMCLI_REPLICATE_CYCLE_DELAY=60	Cycle through the first copy, then wait 60 minutes (delay), and then another cycle, delay, etc.

Start the replicate session with the basic parameters set and run `symreplicate query` to monitor session progress, noting the timing results of the initial copies. Then, adjust the various timing parameters to best accommodate the copy requirements for your needs.

Replication Log Entries

You can track the steps in a `symreplicate` session by setting the `SYMCLI_REPLICATE_LOG_STEP` entry in the options file to `TRUE`. This option causes `symreplicate` to write 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.

Refer to *Symreplicate File Parameters* on page 3-49 for options file syntax.

Clustered SRDF/AR Environments

Since Enginuity 5669, Symmetrix arrays support clustered SRDF/AR environments for multiple node (host) capability. Clustered SRDF/AR provides the capability to start, stop, and restart replication sessions from any host connected to any local Symmetrix array participating in the replication session.

The clustered SRDF/AR environment allows the replication log file to be written 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 Symmetrix array containing the log file would then be able to restart the SRDF/AR session from where it left off.

To write the log file to the SFS, you must specify the ID of the Symmetrix array (`-sid`) where the log file is to be stored at the start of the replication session, along with a group name (`-g`, `-cg`) and an optional user log filename (`-log`). For example:

```
symreplicate start -g session1 -log srdfar1.log -sid 201
```

Note: Not specifying the Symmetrix ID (`-sid`) at the start of the session, causes the log file to be written to local disk using the default SYMAPI log directory, which is not restartable from another node.

If you begin a session and specify a user log file name (`-log`), you must specify the `-log` option for all other commands in the session sequence.

If you begin a session specifying only the group name (`-g`, `-cg`), the log file will be named the same as the group, and must be specified using only the `-g` option for all other commands in the session sequence.

If the primary node fails at some point in the replication process, the SRDF/AR session can now be restarted from another local host using the following command:

```
symreplicate restart -g session1 -log srdfar1.log -sid
201 -recover
```

The `-recover` option is used to recover the device locks from the previously started session.

Note: Before using the `-recover` option, you must ensure that no other replication session using the same devices is currently running.

Note: The device group does not need to be defined on the restart node (host).

You can display a list of the current SRDF/AR log files that have been written to the SFS by using the `list` command with the `-sid` option as follows:

```
symreplicate list -sid 201
```

By including the `-sort` option with the `list` command, you can sort the log file list by name (default) or type.

To display the information content of a particular log file using the `show` command, you must specify the log filename (`-log`) and the Symmetrix ID (`-sid`) as follows:

```
symreplicate show -log srdfar1.log -sid 201 -all
```

The `-all` option (default) is used to display all available information contained in the log, including command-line arguments (`-args`), devices (`-devs`), and options (`-opts`). Refer to the *EMC Solutions Enabler Symmetrix Command Reference* manual for a description of each option.

To delete a particular log file written to the SFS, you must specify either the group name (`-g`) or the log filename (`-log`) (depending on if you defined a user log name when you began the session) with the `delete` command as follows:

```
symreplicate delete -log srdfar1.log
```

Setting Replication Retry and Sleep Times

You can control how long and how often `symreplicate` executes certain control operations by setting the following parameters in the `symreplicate` options file as follows:

```
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 RDF 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 RDF 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 RDF device state.

TimeLimit indicates how long `symreplicate` will retry certain types of operations.

SleepTime indicates the minimum time `symreplicate` should sleep before retrying certain operations.

Factor is used to indicate the maximum time that `symreplicate` sleeps before checking again if devices are in a specific state.

Refer to *Symreplicate File Parameters* on page 3-49 for expanded options file descriptions and syntax.

Note: On restart, if you specify an options file, the following options may not be changed: `SYMCLI_REPLICATE_USE_FINAL_BCV` or `SYMCLI_REPLICATE_HOP_TYPE`. If attempted, an error message is displayed. All other options may be specified and any new values take effect immediately.

Symreplicate File Parameters

The `symreplicate` file is where you can set and edit required parameter entry lines to control the replicate behavior. The following are possible parameter entries and values for the options file:

`SYMCLI_REPLICATE_HOP_TYPE=RepType`

Defines your configured environment in which to operate the data replication 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 Symmetrix array (for multi-hop only) with a replicate data copy (TRUE is the default). If the option is set to FALSE, the second hop BCV devices will be omitted.

`SYMCLI_REPLICATE_PROTECT_BCVS=`

NONE | BOTH | LOCAL | REMOTE | FIRST_HOP | SECOND_HOP

By default (NONE), establishes BCV-STD pairs without the protective establish behavior, relating to two-way mirrored BCV devices. When set to LOCAL or REMOTE, causes the two mirrors of the BCV to be moved or joined to the standard device. When set to BOTH, both the local BCV mirrors and the remote BCV mirrors get joined to their standard device. When set to FIRST_HOP or SECOND_HOP performs the protect BCV establish for first or second hop devices only in a multi-hop configuration.

`SYMCLI_REPLICATE_CYCLE=CycleTime`

Defines the period to wait between copy operations in total *minutes* or in an *hours:minutes* (hh:mm) format.

`SYMCLI_REPLICATE_CYCLE_DELAY=Delay`

Specifies the minimum time to wait between adjacent cycles. Even if a cycle overruns the specified *CycleTime* and *OvfMethod* is set to IMMEDIATE when *Delay* is specified, the session waits this delay time before beginning another cycle.

`SYMCLI_REPLICATE_NUM_CYCLES=NumCycles`

Specifies the number of cycles to perform before exiting. If you specify a value of zero, the replicate session cycles forever. The *NumCycles* default value is zero.

`SYMCLI_REPLICATE_CYCLE_OVERFLOW=OvfMethod`

Describes what to do if the cycle overruns the specified *CycleTime*. Possible *OvfMethod* values are:

IMMEDIATE — Begins next cycle immediately (the default)

NEXT — Skips this copy cycle and wait for the next to begin

`SYMCLI_REPLICATE_LOG_STEP=TRUE | FALSE`

When set to 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 Symmetrix array lock). Currently, the general *TimeLimit* only applies when initiating an RDF split or establish operation. The default general *TimeLimit* is 00:30 if not specified.

The *TimeLimit* value enables you to control how long `symreplicate` retries certain types of operations. *TimeLimit* must be specified using one of the following formats:

hh:mm — Specifies the number of hours and minutes

sss — Specifies the number of seconds

A *TimeLimit* specified as zero (0) indicates that no time limit applies, causing the operation to be retried indefinitely.

`SYMCLI_REPLICATE_RDF_TIME_LIMIT=<TimeLimit>`

Indicates how long to wait for RDF 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 RDF *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 Symmetrix array lock). Currently, the general *SleepTime* only applies when initiating an RDF split or establish operation. The default general *SleepTime* is 10 seconds if not specified.

The *SleepTime* value enables you to control how long `symreplicate` sleeps before retrying certain types of operations. *SleepTime* must be specified using one of the following formats:

hh:mm — Specifies the number of hours and minutes

sss — Specifies the number of seconds

A *SleepTime* must be specified as greater than zero (0).

`SYMCLI_REPLICATE_RDF_SLEEP_TIME=<SleepTime>`

Indicates the minimum length of time that `symreplicate` should sleep before retrying an RDF 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 RDF *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 RDF devices have entered a specific state. The product of this value multiplied by the sleep time gives the maximum time that `symreplicate` sleeps. The factor is specified using a nonzero integer. If not specified, the default factor is 4.

By default, `symreplicate` sleeps between 15 and 60 seconds when checking on the state of RDF devices, up to a maximum time of 4 hours.

`SYMCLI_REPLICATE_TF_CLONE_EMULATION=<TRUE|FALSE>`

Indicates that TimeFinder clone emulation is enabled. The TimeFinder clone emulation default is `FALSE` (disabled). A value of `TRUE` indicates that clone emulation is enabled.

`SYMCLI_REPLICATE_PERSISTENT_LOCKS=<TRUE|FALSE>`

Allows device locks to remain persistent in the event of a system crash or component failure. When set to `TRUE`, causes `symreplicate` to acquire the device locks for the replication session with the `SYMAPI_DLOCK_FLAG_PERSISTENT` attribute. When set to `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.

Option File Format

The options file should conform to the following syntax example, where 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>
```

Note that for proper session behavior, either a *CycleTime* or a *Delay* time nonzero value should be specified, even though their default values are zero. The *RepType* must be specified.

Relative Symmetrix Manager Functions

If you need to convert existing command scripts generated for previous versions of the Symmetrix Manager (now called EMC ControlCenter™) to the SYMCLI command set, refer to Table 3-3, which lists the Symmetrix Manager SM-CLI functions with equivalent SYMCLI commands.

Table 3-3 Symmetrix Manager Relationship

SM-CLI Functions	Equivalent SYMCLI Commands
Target takeover	<code>symrdf -g DgName failover</code>
Source takeover	<code>symrdf -g DgName failback</code>
Pre-Processing	<code>symrdf -g DgName split</code>
Save source data	<code>symrdf -g DgName establish</code>
Save target data	<code>symrdf -g DgName restore</code>
Source preparation	<code>symrdf -g DgName update</code>
Copy source to target	<code>symrdf -g DgName -full establish</code>
Copy target to source	<code>symrdf -g DgName -full restore</code>
Suspend links	<code>symrdf -g DgName suspend</code>
Resume links	<code>symrdf -g DgName resume</code>

RDF Consistency Group Operations

An *RDF consistency group* is a composite group comprised of Symmetrix RDF devices (RDF1 or RDF2), which has been enabled for remote database consistency. The devices in the consistency group are specially configured to act in unison to maintain the integrity of a database when distributed across multiple Symmetrix arrays or across multiple devices within an array. Prior to Solutions Enabler version 6.0, consistency protection for RDF devices required PowerPath support. Beginning with Solutions Enabler version 6.0 and Enginuity 5671, consistency protection for SRDF/Asynchronous devices is provided using Multi Session Consistency (MSC).

Note: PowerPath consistency group operations requires Enginuity 5265 or higher, PowerPath version 1.5 or higher, and an SRDF/CG license.

Note: RDF consistency group operations for SRDF/A using MSC requires Enginuity 5671 or higher and an SRDF/CG license.

For information on consistency protection of SRDF/A devices, refer to *Multi Session Consistency (MSC) Protection for SRDF/A* on page 3-57.

Composite groups are initially created using the `symcg create` command and then populated with devices. To be enabled as an RDF consistency group, the group must be defined as a type RDF1 or RDF2 and include a consistency protection option (`-ppath` or `-rdf_consistency`). Refer to *Creating a Composite Group* on page 3-61 for instructions.

Note: Another way to ensure the integrity of a remote database is to use Domino modes (refer to section *Domino Effect On* on page 2-55).

PowerPath Consistency Protection

If one or more source (R1) devices in a PowerPath enabled RDF consistency group cannot propagate data to their corresponding target (R2) devices, the PowerPath-connected devices suspend all the source (R1) propagation from these devices in the RDF consistency group. This ensures that all data flow to the consistency group's target (R2) side is instantly and completely halted (see Figure 3-15 on page 3-56), and that a consistent database (up to the point in time of data propagation failure) exists on the remote side of the configuration. This ensures the integrity of the remote database.

Note: PowerPath enabled RDF consistency group operations do not presently support concurrent RDF devices.

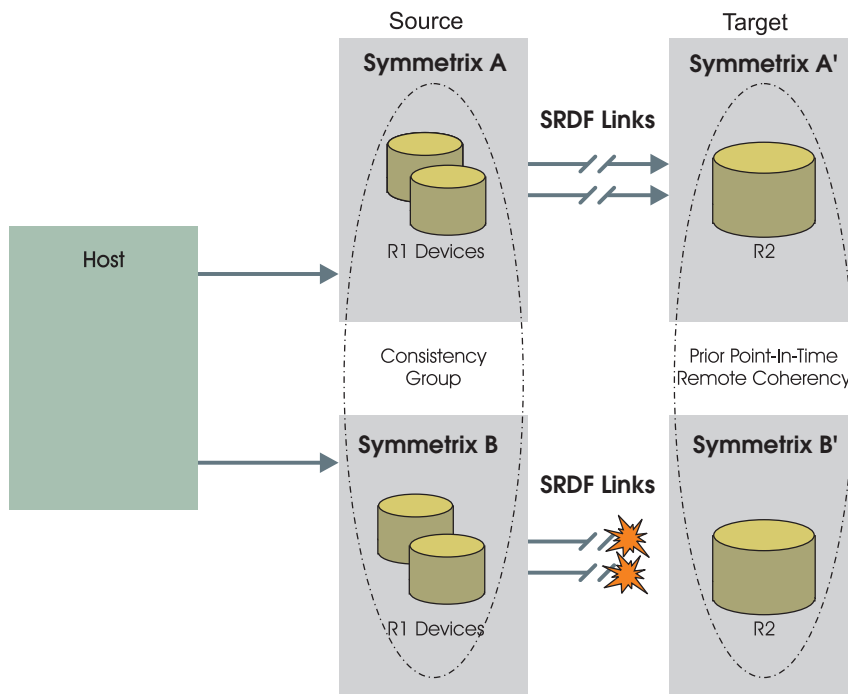


Figure 3-15 All Data Propagation is Suspended Upon Any Link Failure

I/O to the local consistency group devices automatically resumes when the RDF source (R1) devices in the consistency group are suspended. While these updates are not immediately sent to the target side, they are propagated through normal SRDF operation once SRDF mirroring is resumed.

PowerPath enabled consistency group operations and their associated PowerPath-connected devices can be managed using the `symcg` command. Refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide* for additional information on composite groups using the `symcg` command.

Note: For additional information, you can refer to the following white paper: *Using SYMCLI to Implement PowerPath Consistency Protection with SRDF Family Products*.

Multi Session Consistency (MSC) Protection for SRDF/A

Since Enginuity 5671, consistency protection for SRDF/Asynchronous devices is provided using Multi Session Consistency (MSC). If one or more source (R1) devices in an SRDF/A MSC enabled RDF consistency group cannot propagate data to their corresponding target (R2) devices, the MSC process suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets

SRDF/A with MSC is supported by an RDF process daemon that performs cycle-switching and cache recovery operations across all SRDF/A sessions in the group. This ensures that a consistent R2 data copy of the database exists at the point-in-time any interruption occurs. A composite group must be created using the RDF consistency protection option (`-rdf_consistency`) and must be enabled using the `symcg enable` command before the RDF daemon begins monitoring and managing the MSC consistency group. Refer to *Creating a Composite Group and Enable/Disable RDF Consistency* on page 3-61 for more information.

At the time of an interruption (SRDF link failure), MSC analyzes the status of all SRDF/A sessions and either commits the last cycle of data to the R2 target or discards it. To prevent impact to the host, the MSC cycle switching process is coordinated among the participating Symmetrix arrays to switch during a brief period of time when no host writes are being serviced.

Note: For additional information, you can refer to the following white paper: *Using SYMCLI to Implement RDF Consistency Protection with SRDF Family Products*.

RDF Daemon Support for MSC

The RDF process daemon maintains consistency for enabled composite groups across multiple Symmetrix arrays for SRDF/A with MSC. For the MSC option (`-rdf_consistency`) to work in an RDF consistency-enabled environment, each locally-attached host performing management operations must run an instance of the RDF daemon (`storrd`). Each host must also be running an instance of the base daemon (`storapid`), which coordinates all Symmetrix locks and parallel application syscalls. Optionally, if the Group Naming Services (GNS) daemon is also running, it will communicate the composite group definitions back to the RDF daemon. If the GNS daemon is not running, the composite group must be defined on each host individually.

Additional data about the current state of a composite group is communicated to the RDF daemon via files written to the Symmetrix file system. MSC requires that the RDF daemon exist and every attempt will be made to start or restart the daemon to perform cycle switching for SRDF/A. Failure to switch SRDF/A cycles may cause all SRDF/A sessions to be dropped due to a full cache slot. If SRDF/A sessions have been dropped, the SYMAPI and RDF daemon logic determines whether to commit or discard the data accumulated in cache memory.

For redundant consistency protection of RDF composite groups, multiple instances of the RDF daemon can be running at the same time on separate hosts. Each host must have a common view of the composite group being monitored. All redundant daemons run simultaneously, monitoring and switching independently of each other. If one of the redundant daemons fails, the other existing daemon(s) will complete the task.

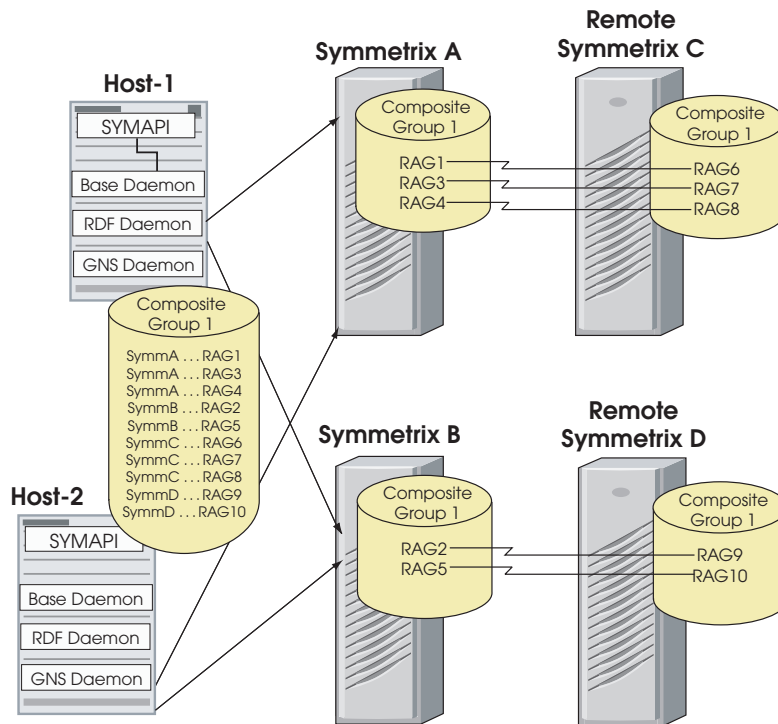


Figure 3-16 RDF Daemon Host Visibility

How to set up the RDF Daemon Option

Use of the RDF daemon is optional and can be enabled or disabled on each host via the SYMAPI options file setting `SYMAPI_USE_RDFD`. This option is enabled as follows:

```
SYMAPI_USE_RDFD=ENABLE
```

The default setting is `DISABLE`. Setting this option to `ENABLE` enables the RDF daemon for any applications using the default SYMAPI configuration database file and SRDF/A MSC.

How to Start and Stop the RDF Daemon

There are three ways that the RDF daemon can be started. If the RDF daemon is enabled, the daemon will be started automatically by the Solutions Enabler libraries the first time they attempt to connect with it, which can cause a slight delay in performance on that initial connection while the daemon starts and builds its cache.

Note: Prior to starting `storrd`, ensure that your default SYMAPI configuration database is up-to-date, since `storrd` uses the information stored in it to establish contact with your Symmetrix arrays.

Alternatively, the daemon can be started manually via the `stord` command line utility as follows:

```
stord start storrd [-wait Seconds]
```

Note: The `stord` command requires a path of `/usr/storapi/storbin`.

By default, the `stord` command waits 30 seconds to verify that the daemon is running. To override this, use the `-wait` option.

In addition, the daemon can be set to start automatically every time the local host is booted using the following command line:

```
stord install storrd -autostart
```

Pre-starting the daemon, either manually or via the automatic option, is useful because the daemon may take a while to initially construct its cache — depending on the number of groups and Symmetrix arrays it has to load.

If the daemon is stopped for some reason, it can optionally be restarted automatically by an internal Solutions Enabler watchdog mechanism. A combination of the watchdog mechanism and the

auto-start option described above can be used to ensure that the daemon is always running.

To stop the RDF daemon, use the following command:

```
stordaeomon shutdown storrdfd|all [-wait Seconds]
```

Applying the `all` option will stop all of the RDF daemons currently running.

RDF Daemon Optional Behavior Parameters

Editing or Removing Parameters

The `storrdfd` file contains a set of parameters that can be modified to affect RDF daemon behavior via SYMCLI or SYMAPI commands.

The file contains editable behavior parameters set to certain optional defaults in the line entries. Commented lines beginning with a pound sign (#) are ignored.

To remove any parameter option, remove the line entry, rename the file, or comment the line by adding a pound sign (#) at the beginning of the line entry.

The following are possible optional parameter entries for the options file:

Table 3-4 RDF Daemon Optional Behavior Parameters

Optional Behavior Parameter	= <OptValue DefaultValue>	Description
storrdfd:autorestart	= enable disable	If set to <code>enable</code> , make use of the watchdog mechanism to automatically restart the daemon if it crashes.
storrdfd:rdfd_num_dedicated_gks	= 0 - 20 2	Reserves a specified number of gatekeeper devices for exclusive use by the RDF daemon.
storrdfd:rdfd_num_based_connections	= 5 - 40 20	Opens a specified number of connections to the base daemon.
storrdfd:rdfd_group_monitor_interval	= 10 - 60 15	Specifies a maximum number of seconds to wait before checking the Symmetrix File System data for updated composite group status.
storrdfd:rdfd_main_interval	= 15 - 120 30	Specifies a maximum number of seconds to wait before checking for Symmetrix configuration changes.
storrdfd:rdfd_db_to_disk_interval	= 30 - 600 120	Specifies a maximum number of seconds to wait before writing the RDF daemon's internal composite group definition database to disk.

Creating a Composite Group

Initially, you must explicitly create an empty composite group (that can be populated with devices) using the `symcg` command. When you create a composite group, you assign it a group name. Creating a composite group does not require the installation of PowerPath unless you want to enable the group for PowerPath consistency.

For example, to create a composite group for PowerPath consistency protection named `mycg1`, enter:

```
symcg create mycg1 -ppath -type rdf1
```

Once you create the group, you can add RDF devices to the group. These RDF devices can belong to different RDF groups within a Symmetrix array and to different Symmetrix arrays.

Note: The composite group must be added to PowerPath (`-ppath`) at the time of creation if you want to enable it for PowerPath consistency protection. Alternatively, you can set the SYMAPI behavior parameter `SYMAPI_RDF_CG_TO_PPATH = ENABLE`.

To create a composite group for MSC consistency protection named `mycg1`, enter:

```
symcg create mycg1 -rdf_consistency -type rdf1
```

Enable/Disable RDF Consistency

You can enable or disable consistency protection for all the standard devices in a composite group. When you enable the composite group for consistency, the group is referred to as an RDF consistency group. This functionality requires SRDF/CG and SRDF/A licenses.

For example, to establish (enable) consistency to group `mycg1`, enter:

```
symcg -cg mycg1 enable
```

To disable consistency to group `mycg1`, enter:

```
symcg -cg mycg1 disable
```

Note: All standard devices in the composite group must be in asynchronous mode to enable for RDF consistency using MSC.

Note: Domino mode must be disabled to enable the RDF consistency state on the device.

Deleting a Composite Group

You can delete a specified composite group. If there are members in the group, the command is rejected unless you use the force (`-force`) option.

For example, to delete a composite group (RDF consistency group) (`mycg1`), regardless if the group has members or not, enter:

```
symcg delete mycg1 -force
```

Note: If the composite group is enabled for MSC RDF consistency protection, the `-symforce` option must be used. The composite group will remain enabled but will be removed from the SYMAPI database.

Note: Refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide* for additional information on composite groups using the `symcg` command.

TimeFinder/Snap and Clone State Reference

This chapter describes the applicable TimeFinder/Snap and TimeFinder Clone pair states and the available SRDF operations.

- ◆ Copy Session Pair StatesA-2
- ◆ TimeFinder/Snap Pair StatesA-2
- ◆ SRDF Operations for TimeFinder/Snap Copy SessionsA-3
- ◆ TimeFinder Clone Pair StatesA-5
- ◆ SRDF Operations for TimeFinder Clone Copy SessionsA-5
- ◆ Setting Snap and Clone Devices to Asynchronous Mode.....A-8

Copy Session Pair States

Certain SRDF operations are not allowed within Symmetrix storage 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 Snap or Clone copy session devices. This chapter describes each of the Snap and Clone pair states and what SRDF operations are available within each state.

TimeFinder/Snap Pair States

Table A-1 provides a description the various TimeFinder/Snap pair states.

Table A-1 TimeFinder/Snap Pair States

State	Description
Created	A virtual copy session between a source device and a target virtual device has been created.
CreateInProg	A virtual copy session between a source device and a target virtual device is in progress.
Not Created	The virtual device is not actively participating in a virtual copy session.
Copied	The entire device has been written to and is in a fully copied state.
CopyOnWrite	Devices participating in the virtual copy session have been activated. Data and pointers will be copied upon any first write to a track on the source device or any host write to the VDEV.
Restored	Data has been fully restored to the specified target device.
RestInProg	Data restore to the specified target device is in progress.
Invalid	Applies to a meta device only. The meta device states do not consistently match.
Failed	The virtual copy session has failed because the log device pool is full.
TermInProg	The virtual copy session between a source device and a target virtual device is in the process of terminating.

SRDF Operations for TimeFinder/Snap Copy Sessions

Table A-2 identifies which SRDF actions are available for use within each of the TimeFinder/Snap pair states. The following notes describe certain Snap pair states to consider when deciphering the table.

- ◆ Target columns are only applicable when the snapshot pair state is RestInProg and apply to the device to which you are restoring the data.
- ◆ If the snapshot pair state is Invalid, no SRDF actions are allowed.
- ◆ If the snapshot pair state is Copied or Terminate in Progress, all SRDF actions are allowed.

Table A-2 SRDF Operations for TimeFinder/Snap Copy Sessions

SRDF Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Establish	X		X ^a	
Incremental Establish	X		X	
Split	X	X	X	X
Restore	X ^a		X	
Incremental Restore	X		X	
Failover	X	X	X	X
Failback	X		X	
Suspend	X	X	X	X
Resume	X		X	
Swap	X ^b		X	
Create pair	X		X	
Delete pair	X	X	X	X
Update R1 Mirror	X		X	
Merge Track Tables	X		X	
RW Disable R2 Mirror			X	X

Table A-2 SRDF Operations for TimeFinder/Snap Copy Sessions (continued)

SRDF Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
RW Enable R1 Mirror	X	X		
RW Enable R2 Mirror			X	X
Write Disable R1 Mirror	X	X		
Write Disable R2 Mirror			X	X
Refresh R1 Mirror	X			
Refresh R2 Mirror			X	
Invalidate R1 Mirror	X	X		
Invalidate R2 Mirror			X	X
Ready R1 Mirror	X	X		
Ready R2 Mirror			X	X
Not Ready R1 Mirror	X	X		
Not Ready R2 Mirror			X	X

- a. The `-force` option must be applied only if the snapshot pair state is CopyOnWrite.
- b. If the R1 snapshot source is in a restored state, the `symrdf swap` command is not allowed.

TimeFinder Clone Pair States

Table A-3 provides a description of the various TimeFinder Clone pair states.

Table A-3 TimeFinder Clone Pair States

State	Description
Created	A Clone copy session between a source device and a target device has been created.
CreateInProg	A Clone copy session between a source device and a target device is in progress.
Copied	The entire device has been written to and is in a fully copied state.
CopyInProg	The device is currently being written to. Applies to the <code>-copy</code> option being used.
CopyOnAccess	Devices participating in the Clone copy session have been activated. Any tracks that have been written to the source or written/read from the target will be copied from the target device.
Invalid	Applies to a meta device only. The meta device states do not consistently match.
TermInProg	The Clone copy session between a source device and a target device is in the process of terminating.

SRDF Operations for TimeFinder Clone Copy Sessions

Table A-4 on page A-6 identifies what SRDF actions are available for use within each of the Clone pair states. The following notes describe certain Clone pair states to consider when deciphering the table.

- ◆ If the Clone pair state is Invalid, no SRDF actions are allowed.
- ◆ If the Clone pair state is Copied or Terminate in Progress, all SRDF actions are allowed.

Table A-4 SRDF Operations for TimeFinder Clone Copy Sessions

SRDF Action	R1 Clone Source	R1 Clone Target	R2 Clone Source	R2 Clone Target
Establish	X	X ^a	X	
Incremental Establish	X	X ^a	X	
Split	X	X	X	X
Restore	X		X	
Incremental Restore	X		X	
Failover	X	X	X	X
Failback	X		X	
Suspend	X	X	X	X
Resume	X	X ^a	X	
Swap	X		X	
Create pair	X		X	
Delete pair	X	X	X	X
Update R1 Mirror	X		X	
Merge Track Tables	X		X	
RW Disable R2 Mirror			X	X
RW Enable R1 Mirror	X	X		
RW Enable R2 Mirror			X	X
Write Disable R1 Mirror	X	X		
Write Disable R2 Mirror			X	X
Refresh R1 Mirror	X			
Refresh R2 Mirror			X	
Invalidate R1 Mirror	X	X		
Invalidate R2 Mirror			X	X

Table A-4 SRDF Operations for TimeFinder Clone Copy Sessions (continued)

SRDF Action	R1 Clone Source	R1 Clone Target	R2 Clone Source	R2 Clone Target
Ready R1 Mirror	X	X		
Ready R2 Mirror			X	X
Not Ready R1 Mirror	X	X		
Not Ready R2 Mirror			X	X

a. Only allowed when the pair state is CopyInProg.

Setting Snap and Clone Devices to Asynchronous Mode

For device pairs employing either TimeFinder/Snap or Clone operations, certain device pairs may not be allowed to be set in asynchronous mode (SRDF/A), depending on the current pair state.

The pair status of the R1 devices will be checked before the set asynchronous operation is allowed to be performed.

Note: If the R2 device is a source or target of a snapshot or clone operation, asynchronous mode will not be allowed.

Table A-5 identifies the applicable R1 pair states that will allow SRDF/A (asynchronous) mode to be set.

Table A-5 Asynchronous for Snap and Clone Sessions

Snap/Clone Pair State	R1 Source	R1 Target
Created	X	X ^a
CreateInProg	X	X ^a
NotCreated	N/A	N/A
Copied	X	X ^a
CopyInProg	X	X ^a
CopyOnAccess	X	N/A
CopyOnWrite	X	X ^a
Restored	X	X ^a
RestInProg	X	X ^a
Invalid	N/A	N/A
Failed	N/A	N/A
TermInProg	X	X ^a

a. Asynchronous mode is not allowed to be set if copy pairs were designated as TimeFinder Clone CopyOnAccess.

A

Adaptive Copy-Disk 2-57
Asynchronous mode 3-9

B

BCV devices 1-7
 remote 3-2
Both sides 3-28
Bypassing locks 2-76

C

Checkpoint
 R2 data committed 2-46
Clustered SRDF/AR 3-45
Composite groups
 creating 3-61
 listing 2-2
Composite operations 2-6
Concurrent RDF operations 3-23
Consistency groups 3-61
Consistency protection
 disable 2-46
 enable 2-45
Consistent split
 auto-replication 3-36
 both sides 3-28
Conventions 1-xiv
createpair 2-28

D

Data mobility 3-23
Data replication

 automated 3-31
Databases
 I/O controls 2-15
deletepair 2-28
delta sets 2-55, 3-9
Dependent Write Consistency 3-14
device file 2-60, 2-84
Device groups
 using 1-7
Device SRDF pairs
 create 2-28
 delete 2-28
Disaster recovery 3-23
Domino Effect 2-55
Dynamic RDF 1-7, 2-67
Dynamic SRDF
 adding groups 3-19
 concurrent pairs 2-63
 delete pair 2-64
 failover 2-70
 group operations 3-19
 modifying groups 3-21
 removing groups 3-22
 viewing swap devices 2-68
Dynamic SRDF Groups 3-19

E

Enterprise TimeFinder consistent split 3-28
Establishing SRDF pairs 2-9, 2-11
External locks
 device 2-7

F

- Fabric 1-6, 3-8
- Failback 2-23
- Failover 2-22
 - Dynamic SRDF 2-70
- Force option 2-77
- Freezing databases 2-15

H

- Hot backup 2-16

I

- Invalid tracks threshold 2-57

J

- JRNL0 attribute 2-54

L

- Labeling 3-20
- Listing composite groups 2-2
- Listing RDF devices 2-2
- Listing RDF groups 2-2
- Locks
 - external device 2-7
- Logical volume attributes
 - Adaptive Copy 2-57
 - Domino Effect 2-55
 - JRNL0 2-54

M

- Merging track tables 2-44
- Modes
 - SRDF 2-54
- Multi Session Consistency 1-8, 3-57
- Multi-Hop
 - configurations 3-2
 - operations 3-2

O

- Offline option 2-78
- Options to symrdf 2-72, 2-75, 2-84
 - composite groups 2-81
 - device file 2-84

P

- Parallel RDF groups parameter 2-8, 3-19, 3-21, 3-22
- Pinging SRDF devices 2-4

Q

- Query SRDF devices 2-4

R

- RA groups
 - adding 3-19, 3-22
 - locking 2-8
 - modifying 3-21
 - topology 3-7
- RDF groups
 - adding 3-19, 3-22
 - locking 2-8
 - modifying 3-21
 - topology 3-7
- RDF process daemon 3-57
- Remote BCV devices 3-2
- Replicate data 3-31
- Restoring SRDF devices 2-16, 2-18

S

- Setting SRDF modes 2-54
- Singular control operations 2-29
- Skew parameter 2-57
- Source (R1) device 1-2, 1-7
- Split
 - Enterprise TimeFinder consistent 3-28
- Splitting SRDF pairs 2-13
- SRDF 1-2
 - operations 2-5
- SRDF Automated Replication facility
 - see also SRDF/AR 3-31
- SRDF configurations
 - bi-directional 1-3
 - uni-directional 1-3
- SRDF control operations
 - failback 2-23
 - failover 2-21
 - full establish 2-9
 - full restore 2-16
 - incremental establish 2-11

- incremental restore 2-18
- invalidate R1 mirror 2-40
- invalidate R2 mirror 2-41
- merge track tables 2-44
- not ready R1 mirror 2-42
- not ready R2 mirror 2-43
- read/write enable R1 mirror 2-34
- read/write enable target (R2) device 2-35
- ready R1 mirror 2-41
- ready R2 mirror 2-42
- refresh R1 mirror 2-38
- refresh R2 mirror 2-39
- resume 2-33
- split 2-13
- suspend 2-32
- swap 2-67
- update R1 mirror 2-25
- write disable source (R1) devices 2-35
- write disable target (R2) device 2-36
- SRDF devices 1-7
 - listing 2-2
 - pinging 2-4
 - querying 2-4
 - viewing 2-2
- SRDF implementations
 - Campus Solution 1-5
 - Extended Distance Solution 1-5
- SRDF modes 2-54
 - adaptive copy 2-56
 - adaptive copy change skew 2-58
 - adaptive copy disk 2-57
 - domino effect 2-55
 - semi synchronous 2-54
 - SRDF/Asynchronous 2-55
 - synchronous 2-54
- SRDF Pair States A-2
- SRDF states
 - verify 2-4
- SRDF/A 3-9
- SRDF/AR 3-31
- SRDF/Asynchronous 3-9
- SRDF/CG 3-61
- States
 - RDF operations 2-50
 - SRDF 2-47
- Swap RDF devices 2-67
- Switched RDF 1-6
 - topology 3-8
- SYMCLI 1-2
- SYMCLI commands
 - symioctl 2-15
 - symrdf 2-9, 2-11, 2-13, 2-16, 2-19, 2-21, 2-23, 2-26, 2-32, 2-33, 2-34, 2-35, 2-36, 2-38, 2-39, 2-40, 2-41, 2-42, 2-43, 2-44, 2-50
- Symmetrix Automated Replication 3-31
- Symmetrix Command Line Interface 1-2
- Symmetrix Ordered Write Processing 3-13
- Symmetrix Remote Data Facility 1-2
- symrdf command options
 - all 2-75
 - bcv 2-75
 - brbcv 2-75
 - bypass locks 2-76
 - concurrent 2-76
 - consistency state 2-77
 - consistent state 2-79
 - count 2-77
 - dynamic 2-76
 - enabled consistency state 2-79
 - failed over state 2-79
 - force 2-77
 - help 2-78
 - interval 2-77
 - list 2-3
 - no echo 2-78
 - no prompt 2-78
 - offline 2-78
 - Partitioned state 2-79
 - R1 2-76
 - R2 2-76
 - rbcv 2-75
 - RDFG 2-76
 - remote 2-78
 - SCSI reservations 2-76
 - SID 2-79
 - split state 2-79
 - suspended and link offline state 2-79
 - suspended state 2-79
 - symforce 2-78
 - synchronized state 2-79
 - until 2-80
 - updated state 2-79
 - UpdateInProg state 2-80
 - valid state 2-80

- verbose 2-80
- symsnap A-3, A-5, A-8
- Synchronization
 - confining 2-8

T

- Target (R2) device 1-2, 1-7
- Thawing databases 2-15
- TimeFinder/CG 3-28
- Track tables 2-44

U

- Until option 2-27
- Updates
 - continuous 2-27
- Updating mirror 2-25