EMC Solutions Enabler
Symmetrix SRDF Family CLI
Version 6.2

PRODUCT GUIDE
P/N 300-000-877
REV A07

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

Published March, 2006

Contents

Appendix A TimeFinder/Snap and Clone State Reference

Appendix B SRDF/Star State Reference

Figures

Tables

Preface

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

Chapter 2, *SRDF Control Operations*, identifies the Symmetrix command line interface actions and specific commands required to manage the Symmetrix Remote Data Facility. It focuses on the various arguments, options, and the applications of certain parameters for the SRDF monitor and control actions.

Chapter 3, *Various Remote Operations*, discusses operation, management, and strategies of the various possible Symmetrix Remote Data Facility configurations and how to perform special operations.

Chapter 4, *SRDF/Star*, focuses on the SRDF/Star configuration, which uses concurrent SRDF/Synchronous and SRDF/Asynchronous links to replicate source data synchronously to a nearby regional site and asynchronously to a distant remote site.

Chapter 5, *Performing SRDF Control Operations*, provides examples of the SRDF control operations used to manage devices within various remote SRDF configurations.

Chapter 6, *Implementing Consistency Protection Using RDF-ECA and RDF-MSC*, provides examples for implementing consistency protection across one or more database management systems within an SRDF configuration using RDF Enginuity Consistency Assist (RDF-ECA) for synchronous mode and RDF Multi Session Consistency (RDF-MSC) for asynchronous mode.

Chapter 7, *Implementing Consistency Protection Using PowerPath*, provides examples for implementing consistency protection across one or more database management systems within an SRDF configuration using PowerPath.

Chapter 8, *Performing SRDF/Automated Replication Operations*, provides examples for replicating data in pre-defined cycles using the SRDF automated replication process.

Chapter 9, *Querying and Verifying with SRDF Commands*, provides examples on using the query and verify operations with SRDF family products.

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.

Appendix B, *SRDF/Star State Reference*, describes the allowable SRDF/Star system states for using the `symstar` command arguments.

Related Documentation

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

- *EMC Solutions Enabler Symmetrix CLI Command Reference*
- *EMC Solutions Enabler Support Matrix*
- *EMC Solutions Enabler Installation Guide*
- *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*
- *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*
- *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide*
- EMC Host Connectivity Guides for *[your operating system]*

The following is an EMC Engineering Technical Note of related interest:

- *Using SYMCLI to Implement SRDF/Star*

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

The following type style conventions in this guide:

Palatino, bold	◆ Boldface text provides extra emphasis and emphasizes warnings, and specifies window names and menu items in text.
Palatino, italic	◆ New terms or unique word usage in text ◆ Applies emphasis in examples and in references to book titles and sections.
`Courier, italic`	◆ Identifies variables in a software syntax (non-literal notation)

`Courier`	◆	A fixed space font identifies files and pathnames, and is used in command line entries, displayed text, or program listings.

	◆	System prompts and displays and specific filenames or complete paths. For example:
		`working root directory [/user/emc]:`
		
		`c:\Program Files\EMC\Symapi\db`

`Courier, bold`	◆	Actual user entry in examples. For example: **`symrdf list`**

Where to Get Help

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

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink website (registration required) at:

http://Powerlink.EMC.com

Technical support — For technical support, go to EMC WebSupport on Powerlink. To open a case on EMC WebSupport, you must be a WebSupport customer. Information about your site configuration and the circumstances under which the problem occurred is required.

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this guide to:

techpub_comments@EMC.com

PART I

Concepts and Procedures

The Concepts and Procedures part of this product guide provides conceptual information and describes how to perform SRDF operations on Symmetrix devices of local and remote sites, using the Symmetrix command line interface (SYMCLI) of the EMC Solutions Enabler software. These concepts and procedures are described in subsequent chapters as follows:

Chapter 1, *Overview*, provides an overview of the Symmetrix Remote Data Facility business continuance solution and describes the various SRDF site configurations used in a Symmetrix storage complex.

Chapter 2, *SRDF Control Operations*, identifies the Symmetrix command line interface actions and specific commands required to manage the Symmetrix Remote Data Facility. It focuses on the various arguments, options, and the applications of certain parameters for the SRDF monitor and control actions.

Chapter 3, *Various Remote Operations*, discusses operation, management, and strategies of the various possible Symmetrix Remote Data Facility configurations and how to perform special operations.

Chapter 4, *SRDF/Star*, focuses on the SRDF/Star configuration, which uses concurrent SRDF/Synchronous and SRDF/Asynchronous links to replicate source data synchronously to a nearby regional site and asynchronously to a distant remote site.

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

1

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

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.

SRDF provides a recovery solution for component or site failures between remotely mirrored devices, as shown in Figure 1-1. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster.

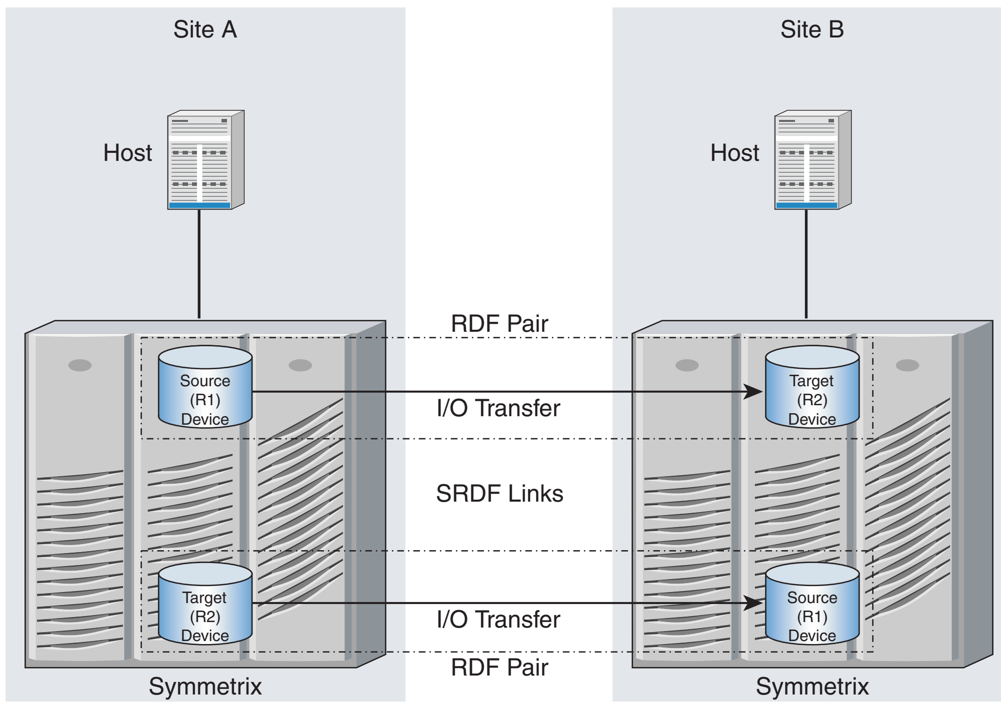

Figure 1-1 SRDF Bidirectional Configuration

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.

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
- SRDF/Star— Disaster Recovery Solution

Note: The terms SRDF and RDF are used throughout this manual and refer to the Symmetrix Remote Data Facility. For a detailed introduction to the Solutions Enabler, SYMCLI, and the Symmetrix array, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

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 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.

SRDF Types of Implementation and Remote Links

Physically, SRDF point-to-point network implementations can be *campus solution* or 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™. The campus solution supports both unidirectional and bidirectional SRDF connections.

SRDF links that remotely connect the Symmetrix sites can transfer data in any of the following protocols (modes):

- Synchronous (SRDF/S)
- Asynchronous (SRDF/A)
- Adaptive Copy

Note: For more information on the available SRDF operational modes, refer to *Setting SRDF Modes* on page 2-57.

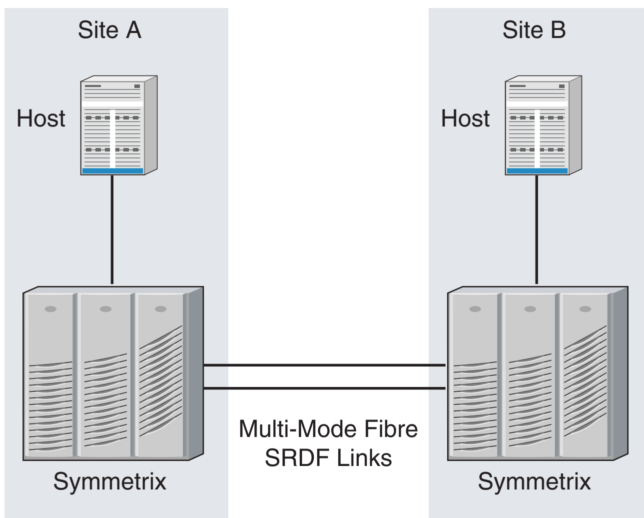

Figure 1-2 SRDF Campus and Distance Business Protection Solutions

---

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.

Depending upon your specific configuration, Symmetrix DMX models can support mixed combinations of port type connectivity, including: Fibre Channel, ESCON, FICON, Gig-E or iSCSI.

Multi-Hop RDF Environments and Automated Replication

An SRDF multi-hop topology allows you to string three Symmetrix sites where (as shown in Figure 1-3) a third RDF site (Site C) is providing business continuance backup to the remote RDF site (Site B). In this multi-hop scheme, Site C is two hops (SRDF links) away, remotely backing up both the production site (Site A) and the remote site (Site B) in the first hop. For more information, refer to Chapter 3, *Various Remote Operations*.

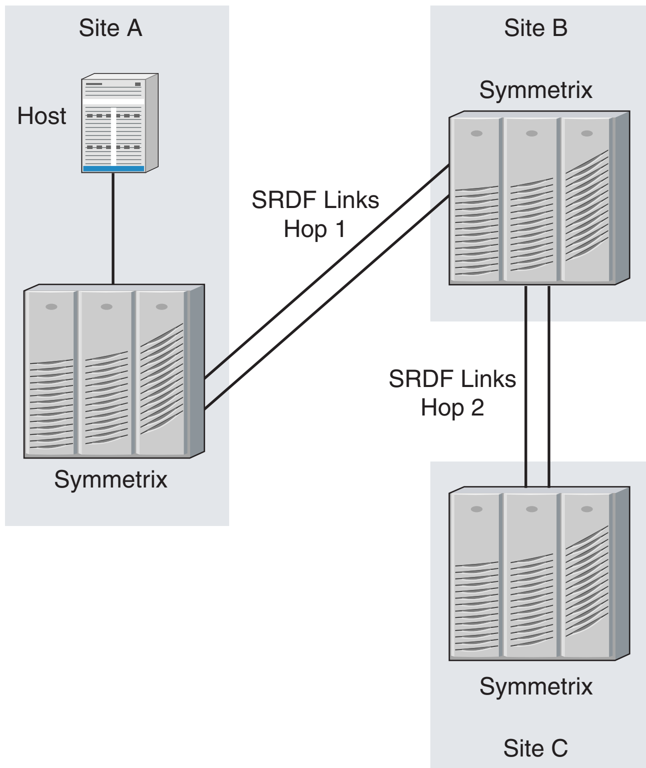

Figure 1-3 SRDF Multi-Hop Solution

In single hop and multi-hop operations you can fully automate backup copying by using the SRDF Automated Replication (SRDF/AR) facility. For more information about automated replication sessions, refer to *SRDF/Automated Replication Operations* on page 3-28.

Switched RDF Environments and Concurrent RDF

An SRDF topology can incorporate open network switching (*fabric*) in the SRDF links (Figure 1-4). 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.

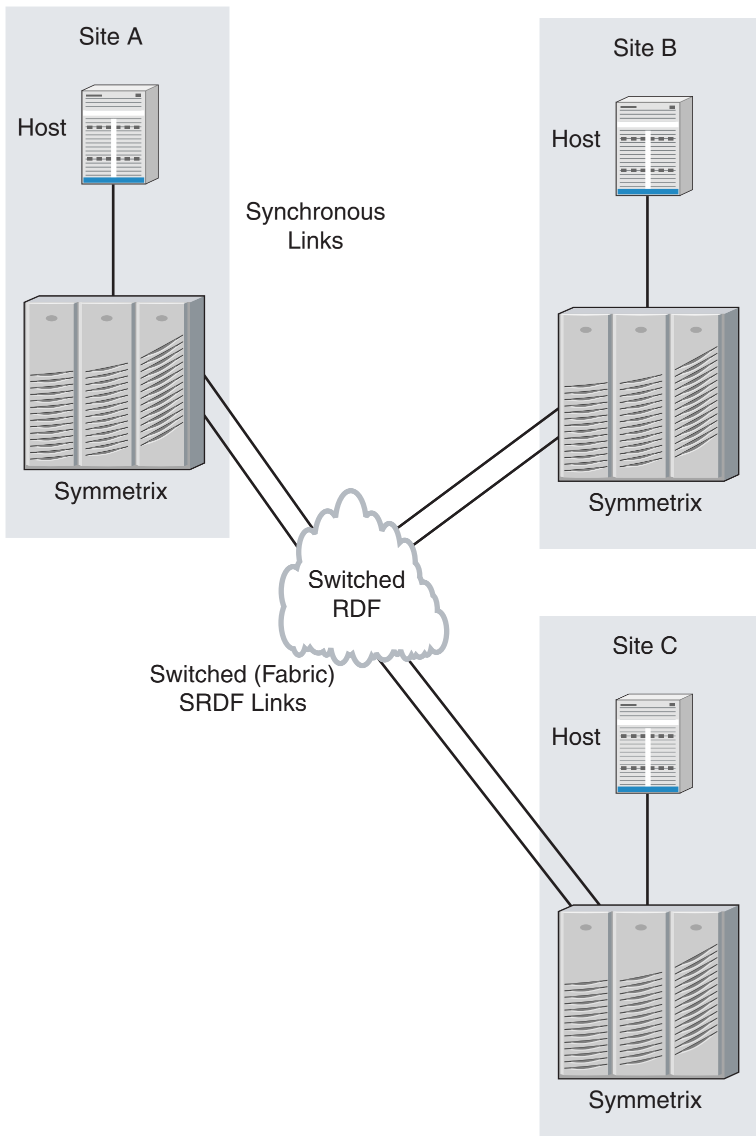

Figure 1-4 Switched (Fabric) RDF Topology

In a Concurrent RDF configuration, a single source (R1) device at site A can concurrently be remotely mirrored to two separate sites (R2 devices) and is supported with ESCON and switched-fibre interfaces. Each of the two remote sites can operate independently, but concurrently, in any of the following SRDF protocols (modes):

- Synchronous (SRDF/S)
- Asynchronous (SRDF/A)
- Adaptive Copy

For more specific information, refer to Chapter 3, *Various Remote Operations*.

SRDF/Star Environment

SRDF/Star is a data protection and failure recovery solution that covers three geographically dispersed data centers in a triangular topology (Figure 1-5). SRDF/Star configures its three sites to protect business data against a primary site failure or a regional disaster, using concurrent RDF capability to mirror the same production data synchronously to one remote site and asynchronously to another remote site:

- The *workload site* of the SRDF/Star topology is the primary data center where the production workload is running.
- The *sync target site* is a secondary site usually located in the same region as the workload site. The production data is mirrored to this site using synchronous replication.
- The *async target site* is a secondary site in a distant location. The production data is mirrored to this site using asynchronous replication.

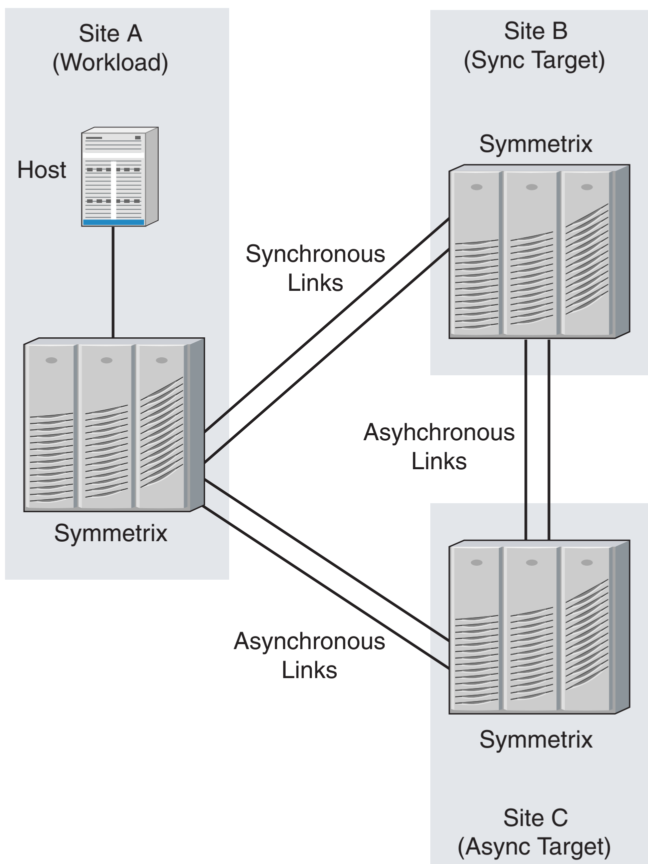

Figure 1-5 SRDF/Star Topology

Note: To perform SRDF/Star operations with Access Control Enabler, you need RDF BASECTRL, BASE, and BCV access types. For more information, refer to *EMC Solutions Enabler Access Control CLI Product Guide*.

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.

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.

Dynamic RDF Devices

Since Enginuity™ Version 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 more information about dynamic SRDF devices, refer to *Dynamic SRDF Pair Operations* on page 2-61.

Dynamic RDF Groups

RDF groups define a collective data transfer and communication path associating and linking the devices of two separate Symmetrix arrays.

Since Enginuity Version 5669, you can dynamically create RDF groups with specified devices on demand while the Symmetrix array is in operation.

For more information about dynamic RDF groups, refer to *Dynamic RDF Group Operations* on page 3-4.

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.

For information on composite groups and using the `symcg` command, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

SRDF Consistency Groups

An S*RDF 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®, Enginuity Consistency Assist 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.

For information on enabling consistency for composite groups, refer to *RDF Consistency Group Operations* on page 3-46

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
`symrdf`	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.
`symreplicate`	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.
`symstar`	The SRDF/Star command performs control operations on a composite group to support a disaster recovery solution. SRDF/Star uses concurrent RDF technology to replicate data from a primary production site to two remote sites. The symstar command provides the following functionality: • An automated setup command • Builds host composite groups • Synchronizes RDF devices and provides consistency protection • Provides switch, cleanup, and resyncrhonization operations between sites

2

SRDF Control Operations

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.

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

Note: Refer to *Listing Devices by Type* on page 2-42 for a description of `symrdf list` options.

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

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

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 Version 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, device groups, and composite 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
```

Preliminary Control Considerations

This section includes information that you should be aware of before using SYMCLI to perform SRDF operations.

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.

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

Migrating Data from R1 to a Larger R2 Device

Beginning with Enginuity Version 5669, you can copy data from an R1 device to a larger R2 device.

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

- Swap or SRDF/Star operations.
- 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.

Disallowing 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.

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.

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

Note: Since Enginuity Version 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

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.

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 (not to be confused with composite groups).

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 in *Composite SRDF Control Operations* on page 2-8. The singular SRDF control operations are described in *Singular SRDF Control Operations* on page 2-26.

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	symrdf Action Arguments	Results
Establishing an RDF pair	`establish -full`	Establishes remote mirroring and initiates a full data copy from the source (R1) device to the target (R2) device.
Incrementally Establishing an RDF pair	`establish`	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	`restore -full`	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	`restore`	Resumes remote mirroring and initiates an incremental data copy from the target (R2) device to the source (R1) device.
Splitting an RDF pair	`split`	Stops remote mirroring between the source (R1) device and the target (R2) device. The target device is made available for local host operations.
Failover	`failover`	Switches data processing from the source (R1) side to the target (R2) side.
Failback	`failback`	Switches data processing from the target side (R2) back to the source (R1) side.
Update R1 mirror	`update`	Updates the source (R1) side with the changes from the target (R2) side while the target (R2) side is still operational to its local host(s).
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.
Delete one-half of an SRDF Pair	`half_deletepair`	Deletes one-half of the designated dynamic RDF pair.
Discard incomplete SRDF/A data	`msc_cleanup`	Initiates a cleanup operation to discard any incomplete SRDF/A data to maintain dependent write consistency.

The SRDF control operations listed in Table 2-1 invoke several singular SRDF control operations, which are listed in Table 2-3 on page 2-27.

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

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 2-6.

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[ile] FileName establish -full
```

Note: For more detail about defining a device file, refer to *Device File* on page 2-49.

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 Array 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.

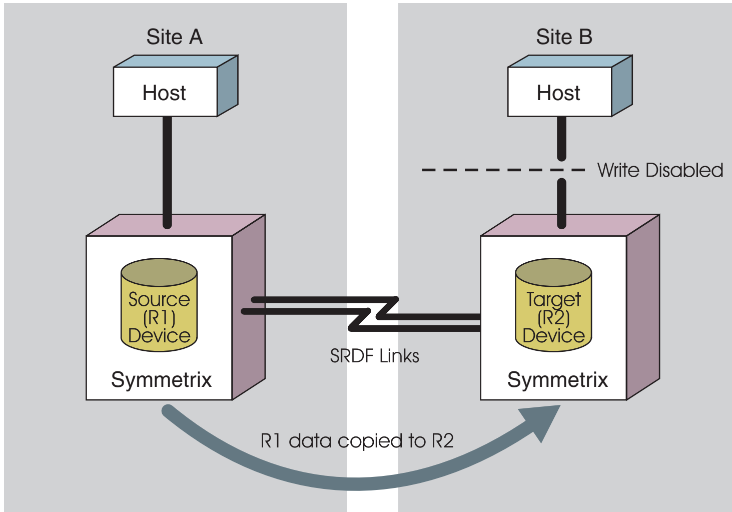

Figure 2-1 Establishing an RDF Pair

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 2-6.

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[ile] FileName establish
```

Note: For more detail about defining a device file, refer to *Device File* on page 2-49.

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 Array 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.

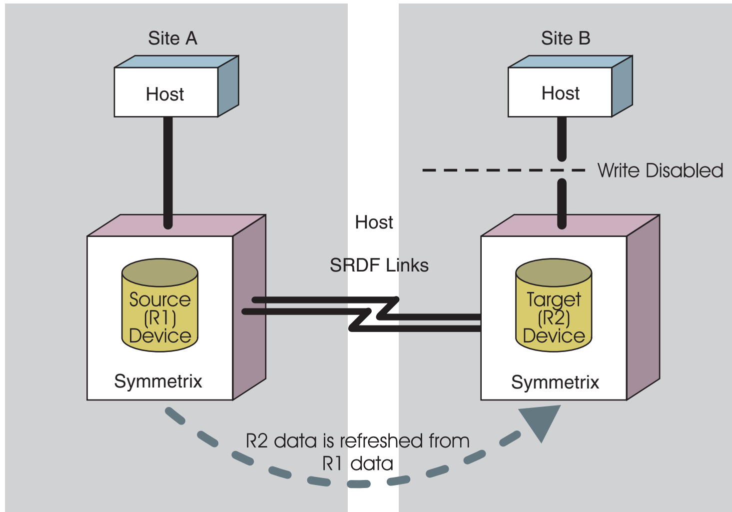

Figure 2-2 Incremental Establish of an RDF Pair

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 2-6.

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[ile] FileName split
```

Note: For more detail about defining a device file, refer to *Device File* on page 2-49.

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
- SyncInProg 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.

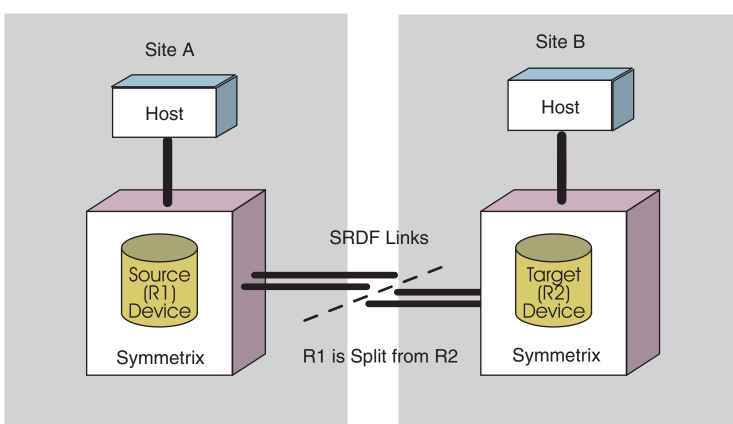

Figure 2-3 Splitting an RDF Pair

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 3-46.

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 2-6.

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[ile] FileName restore -full
```

Note: For more detail about defining a device file, refer to *Device File* on page 2-49.

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 Array 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.

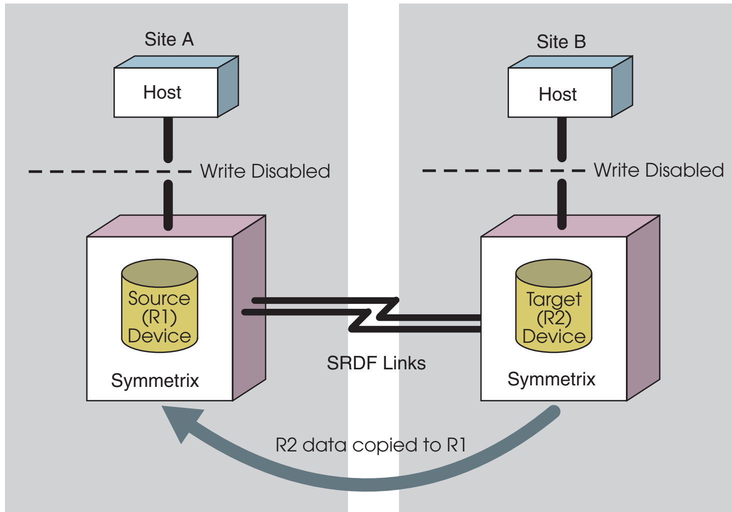

Figure 2-4 Restoring an SRDF Device

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 2-6.

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[ile] FileName restore
```

Note: For more detail about defining a device file, refer to *Device File* on page 2-49.

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 Array Management CLI Product Guide*.

Figure 2-5 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.

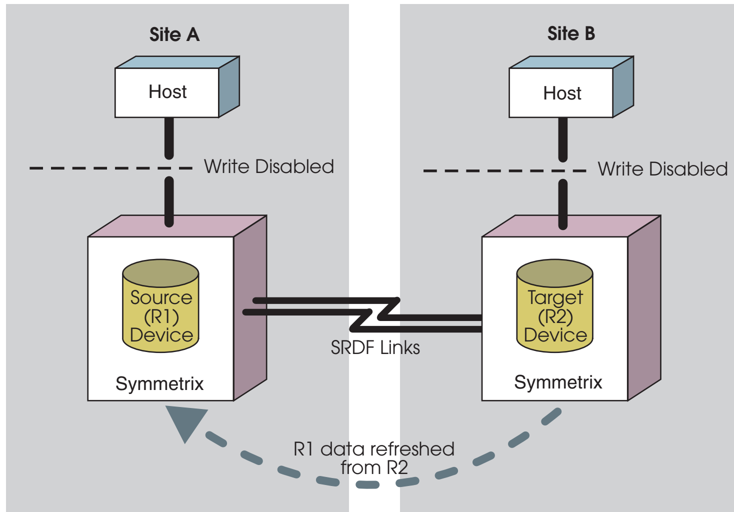

Figure 2-5 Incrementally Restoring an SRDF Device

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 2-20 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 2-6.

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[ile] 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
- SyncInProg
- R1 UpdInProg
- 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.

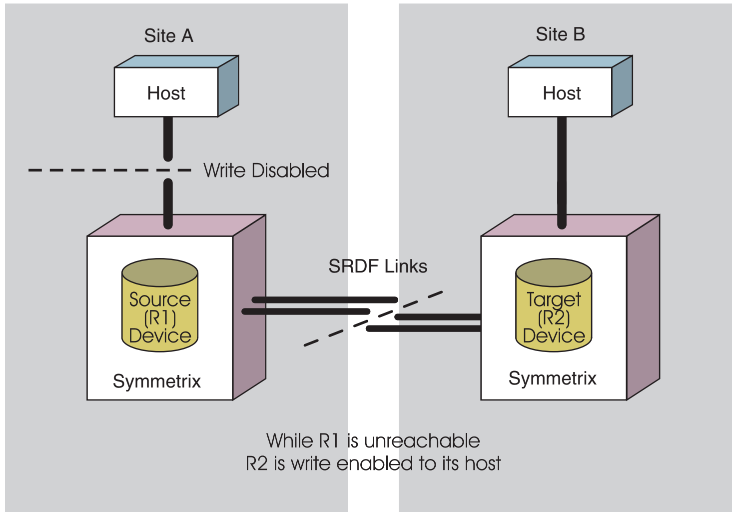

Figure 2-6 Failover of an SRDF Device

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 2-6.

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[ile] 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 Array 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.

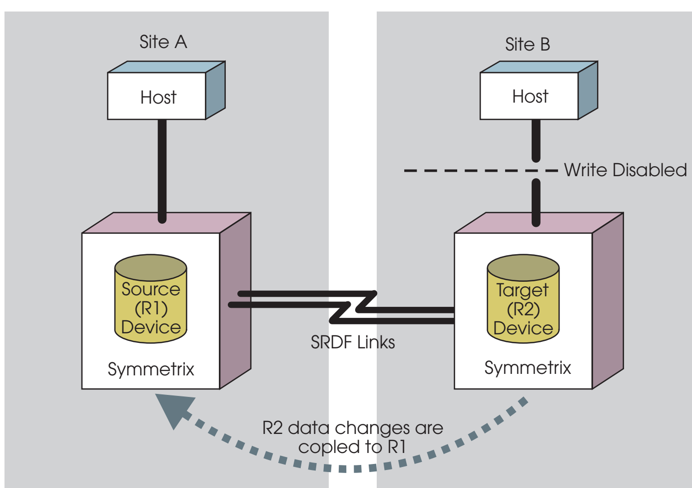

Figure 2-7 Failback of an SRDF Device

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 2-6.

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[ile] 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.

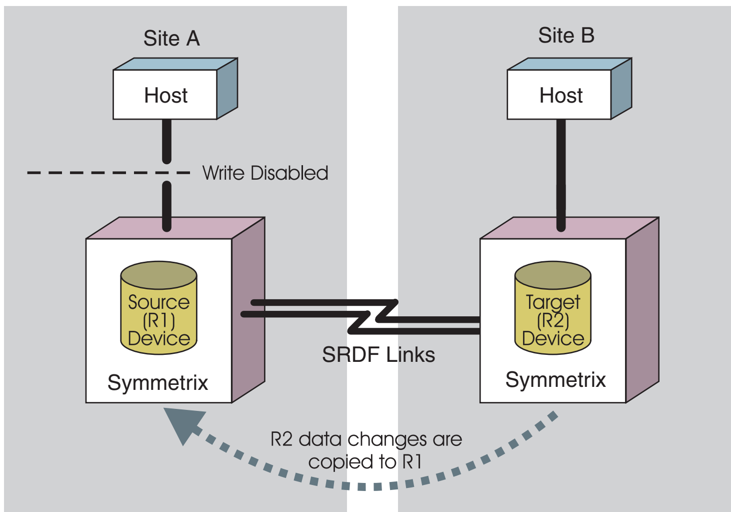

Figure 2-8 Update of SRDF Device Track Tables

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 2-61.

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 2-65.

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.

Delete One-half of an SRDF Pair

Since Solutions Enabler Version 6.1 and Enginuity Version 5671, you can delete one-half of a designated SRDF pair as specified in a device file or by device group. The command cancels the dynamic SRDF pairing information and converts one-half of the specified device pairs from RDF to regular devices. If specified by device file, the devices listed in the first column of the file will be converted to non RDF devices.

For example, to remove the SRDF pairing of RDF group 10 and convert one-half of those paired devices to regular (non RDF) devices, enter:

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

For more information, refer to *Delete One-half of an SRDF Pair* on page 2-65.

Cleanup Incomplete SRDF/A Data

Since Solutions Enabler Version 6.1 and Enginuity Version 5671, the `msc_cleanup` command can be issued for devices operating in SRDF/A mode that have consistency enabled for Multi Session Consistency (MSC). The command may be necessary in certain fault scenarios where all delta sets of a transistion have not been fully applied or discarded. The command can be executed by composite group from the R1 or R2 site or by RDF group from the R2 site. The command maintains dependent write consistency by discarding any incomplete data and committing completed data to the R2 site.

To perform a clean up operation on a composite group (`mycg`) operating in SRDF/A mode, enter:

```
symrdf -cg mycg msc_cleanup
```

For more information on when to execute this command, refer to *Using the msc_cleanup Command* on page 3-48.

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 2-8 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
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 `-force` 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 `-force` 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

Synchronizing Changed 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	symrdf Action Arguments	Results
Suspend SRDF links	`suspend`	Suspends I/O traffic on the SRDF links for the remotely mirrored RDF pair(s) in the group.
Resume SRDF links	`resume`	Resumes I/O traffic on the SRDF links for the remotely mirrored RDF pair(s) in the group.
Write Enable source device	`rw_enable r1`	Write Enables the source (R1) device to its local host.
Write Enable target device	`rw_enable r2`	Write Enables the target (R2) device to its local host.
Write Disable source device	`write_disable r1`	Write Disables the source (R1) device to its local host.
Write Disable target device	`write_disable r2`	Write Disables the target (R2) device to its local host.
Read/Write Disable target device	`rw_disable r2`	Read/Write Disables the target (R2) device to its local host.
Refresh R1 mirror	`refresh r1`	Marks any changed tracks on the source (R1) side to be refreshed from the R2 side.
Refresh R2 mirror	`refresh r2`	Marks any changed tracks on the target (R2) side to be refreshed from the R1 side.
Invalidate R1 mirror	`invalidate r1`	Invalidates all tracks on the source (R1) side so that they can be copied over from the target (R2) side.
Invalidate R2 mirror	`invalidate r2`	Invalidates all tracks on the target (R2) side so that they can be copied over from the source (R1) side.
Make Ready the R1 mirror	`ready r1`	Sets the source (R1) device to be RDF Ready to its local host.

Table 2-3 Singular SRDF Control Operations (continued)

Control Operation	symrdf Action Arguments	Results
Make Ready the R2 mirror	`ready r2`	Sets the target (R2) device to be RDF Ready to its local host.
Make the R1 mirror Not Ready	`not_ready r1`	Sets the source (R1) device to be RDF Not Ready to its local host.
Make the R2 mirror Not Ready	`not_ready r2`	Sets the target (R2) device to be RDF Not Ready to its local host.
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.

Suspending I/O on Links

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[ile] 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:
  - SyncInProg state
  - UpdInProg 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

Resuming I/O on Links

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[ile] 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.

Enabling R1 Writes

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[ile] 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
- SyncInProg
- 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.

Enabling R2 Writes

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[ile] 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

Disabling R1 Writes

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[ile] 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
- SyncInProg
- Suspended
- Split
- Invalid

Disabling R2 Writes

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[ile] 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
  - SyncInProg
- If the device pair(s) are in one of the following states without specifying the `-force` option:
  - Split
  - Failed Over
  - R1 Updated
  - R1UpdInProg
  - Invalid

Disabling R2 Read/Writes

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[ile] 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

Refreshing R1 From the R2

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[ile] 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.

Refreshing R2 From the R1

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[ile] 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.

Invalidating R1 Tracks

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[ile] 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.

Invalidating R2 Tracks

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[ile] 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.

Setting the R1 Ready

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[ile] 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.

Setting the R2 Ready

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[ile] 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.

Setting the R1 Not Ready

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[ile] 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

Setting the R2 Not Ready

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[ile] 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
- SyncInProg
- Suspended
- Partitioned and you are invoking the action from the target side
- Invalid

Merging 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[ile] 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.

Enabling Consistency Protection with SRDF/A

The *enable* action enables consistency protection for devices in SRDF/Asynchronous mode 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 *RDF Consistency Group Operations* on page 3-46.

Disabling Consistency Protection with SRDF/A

The *disable* action disables consistency protection for devices in SRDF/Asynchronous mode 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
```

Command Options with Device Groups

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 2-4 lists the `symrdf` control operations type arguments and options available when operating on RDF device(s) of a specified device group.

Table 2-4 symrdf -g Control Arguments and Possible Options

	Possible Options												
Argument Action	**-force -sym force**	**-by pass**	**-bcv**	**-brbcv -rbcv**	**-all**	**-rdfg**	**-re mote**	**-imme diate**	**-i/-c**	**-v**	**-no echo**	**-no prompt**	**-star**
`deletepair`	✓	✓				✓			✓	✓	✓	✓	✓
`disable`	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓
`enable`	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓
`establish [-full]`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`failback`	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
`failover [-establish]`	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
`invalidate r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`merge`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`not_ready r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`ready r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`refresh r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`restore [-full]`	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
`resume`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`rw_disable r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`rw_enable r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`set mode|domino|skew|nr`	✓	✓	✓	✓	✓	✓			✓	✓		✓	✓
`split`	✓	✓	✓	✓	✓	✓		X	✓	✓	✓	✓	✓
`suspend`	✓	✓	✓	✓	✓	✓		X	✓	✓	✓	✓	✓
`swap`[a]	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
`update [-until]`	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
`write_disable r1 | r2`	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓

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 2-5 lists the `symrdf` view action type arguments and options available when viewing RDF device(s) of a specified device group.

Note: For expanded operational examples using the `symrdf query` and `verify` commands, refer to Chapter 9, *Querying and Verifying with SRDF Commands*.

Table 2-5 symrdf -g View Arguments and Possible Options

	Argument Action		
Options	**query**	**verify**	**checkpoint**
`-all`	✓	✓	✓
`-bcv`	✓	✓	✓
`-consistent`		✓	
`-failedover`		✓	
`-i`, `-c`	✓	✓	✓
`-offline`	✓	✓	✓
`-partitioned`		✓	
`-rbcv`, `-brbcv`	✓	✓	✓
`-rdfa`	✓		
`-rdfg`	✓	✓	✓
`-split`		✓	
`-susp_offline`		✓	
`-suspended`, `-enabled`		✓	
`-synchronized`		✓	
`-syncinprog`		✓	
`-updated`		✓	
`-updateinprog`		✓	
`-valid`		✓	

Table 2-6 lists the `symrdf` view action type arguments and options available when viewing RDF device(s).

Table 2-6 symrdf -g View RDF Device Arguments and Possible Options

	Argument Action	
Options	**list**	**ping**
`-bcv`	✓	
`-concurrent`	✓	
`-consistency`	✓	
`-dup_pair`	✓	
`-dynamic`	✓	
`-half_pair`	✓	
`-i, -c`	✓	✓
`-nobcv`	✓	
`-offline`	✓	
`-R1, -R2`	✓	
`-rdf`		✓
`-rdfa`	✓	
`-rdfg`	✓	
`-resv`	✓	
`-sid`	✓	✓
`-star_async_target`	✓	
`-star_mode`	✓	
`-star_sync_target`	✓	
`-v`	✓	

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.

symrdf -star option

The `symrdf -star` option must be included in the command line for any `symrdf` action argument targeting a device that is currently in SRDF/Star mode.

Note: The `symrdf` command should not be used to alter an SRDF/Star environment. Refer to Chapter 4, *SRDF/Star* for instructions on using the `symstar` command.

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 by Type

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 Enginuity Version 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.

Half Pairs

`The -half_pair` option allows you to list any SRDF devices that are not paired with another device.

For example, to list all of the half pair devices in Symmetrix array 333, enter:

```
symrdf -sid 333 -halfpair list
```

Note: Existing half pair devices could result from an SRDF/Star failover scenario, a `half_deletepair` operation, or a configuration change.

Duplicate Pairs

`The -dup_pair` option allows you to list any SRDF devices that are paired with the same RDF type.

For example, to list all of the duplicate pair devices in Symmetrix array 333, enter:

```
symrdf -sid 333 -dup_pair list
```

Note: Existing duplicate pair devices could result from an SRDF/Star failover scenario or a configuration change.

SRDF/Star Devices

The following list commands are provided to identify which devices are operating in SRDF/Star mode. The `symrdf list -star_mode` command lists all devices that are currently in SRDF/Star mode.

The `-star_async_target` option allows you to list all devices that are asynchronous R2 target devices.

The `-star_sync_target` option allows you to list all devices that are synchronous R2 target 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 redisplays, 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 2-13 on page 2-54.

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

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 (SyncInProg 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 2-24 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 *Using the Immediate Option* on page 3-13 for more information.

Command Options with Composite Groups

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 2-7 lists the `symrdf` control operations type arguments and options available when operating on RDF device(s) of a specified composite group.

Table 2-7 symrdf -cg Control Arguments and Possible Options

	Possible Options									
Argument Action	-force -sym force	-by pass	-bcv	-brbcv -rbcv	-rdfg	-i/-c	-v	-no echo	-no prompt	-star
`establish [-full]`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`failback`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`failover [-establish]`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`invalidate r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`merge`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`not_ready r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`ready r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`refresh r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`restore [-full]`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`resume`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`rw_enable r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`set mode|domino|skew|nr`		✓	✓	✓	✓	✓	✓	✓	✓	✓
`split`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`suspend`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`update [-until]`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
`write_disable r1 | r2`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

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 2-8 lists the `symrdf` view action type arguments and options available when viewing RDF device(s) of a specified composite group.

Note: For expanded operational examples using the `symrdf query` and `verify` commands, refer to Chapter 9, *Querying and Verifying with SRDF Commands*.

Table 2-8 symrdf -cg View Arguments and Possible Options

	Argument Action		
Options	**query**	**verify**	**checkpoint**
`-bcv`	✓	✓	
`-consistent`		✓	
`-cg_consistent`		✓	
`-detail`	✓		
`-failedover`		✓	
`-i, -c`	✓	✓	✓
`-offline`	✓	✓	✓
`-partitioned`		✓	
`-split`		✓	
`-susp_offline`		✓	
`-suspended, -enabled`		✓	
`-synchronized`		✓	
`-syncinprog`		✓	
`-updated`		✓	
`-updateinprog`		✓	
`-valid`		✓	

For a brief description of all the various options in these tables, refer to *Targeting All Devices* on page 2-41 where the option descriptions begin.

Command Options with Device Files

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

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

	Possible Options											
Argument Action	**-sid**	**-type**	**-force -sym force**	**-by pass**	**-rdfg**	**-re mote**	**-imme diate**	**-i/-c**	**-v**	**-no echo**	**-no prompt**	**-star**
`createpair [-g Name] -invalidate|-establish|-restore`	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
`deletepair`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`disable`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`enable`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`establish [-full]`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`failback`	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓
`failover [-establish]`	✓		✓	✓	✓		X	✓	✓	✓	✓	✓
`invalidate r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`merge`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`not_ready r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`ready r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`refresh r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`restore [-full]`	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓
`resume`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`rw_enable r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓
`rw_disable r1 | r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓

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

	Possible Options											
Argument Action	**-sid**	**-type**	**-force -sym force**	**-by pass**	**-rdfg**	**-re mote**	**-imme diate**	**-i/-c**	**-v**	**-no echo**	**-no prompt**	**-star**
`set mode|domino|skew`	✓			✓	✓			✓	✓		✓	✓
`split`	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓
`suspend`	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓
`swap`[a]	✓		✓	✓	✓			✓	✓	✓	✓	✓
`write_disable r1	r2`	✓		✓	✓	✓			✓	✓	✓	✓	✓

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

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

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

For a brief description of all the various options in these tables, refer to *Targeting All Devices* on page 2-41 where the option descriptions begin.

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.

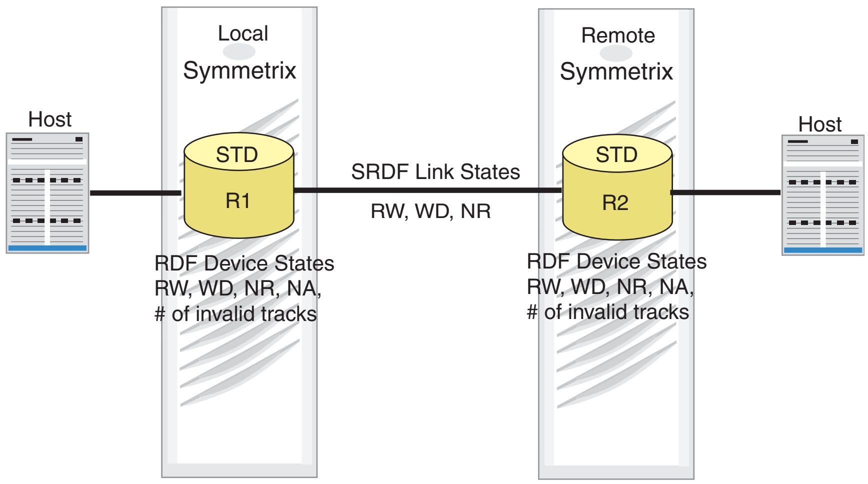

Figure 2-9 RDF Pair and Link States

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 2-11 provides a description of the various SRDF pair states.

Table 2-11 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.

Table 2-11 SRDF Pair States (continued)

State	Description
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 2-12 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 2-12:

Table Legend

Not Ready: Disabled for both reads and writes

Ready: Enabled for both reads and writes

WD: Write Disabled

Table 2-12 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 2-13 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 2-13 as ✓**g**).

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

Table 2-13 RDF Control Operations and Applicable States

Control Operation	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned	R1 Updated	R1 UpdInProg	Invalid	Consistent
establish -full			✓	✓[a]	✓[m]				✓[j]	
split	✓[n]	✓[i]		✓			✓			✓
establish			✓	✓					✓[j]	
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]	✓		✓			

Table 2-13 RDF Control Operations and Applicable States (continued)

Control Operation	SyncInProg	Synchronized	Split	Suspended	Failed Over	Partitioned	R1 Updated	R1 UpdInProg	Invalid	Consistent
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]	✓
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]	✓	✓		✓
half_deletepair			✓	✓	✓	✓				
msc_cleanup			✓	✓	✓	✓[h]				

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[ile] 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
```

Note: Beginning with Enginuity 5771, semi-synchronous mode is no longer supported.

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. For details, refer to Appendix A, *TimeFinder/Snap and Clone State Reference*.

For additional information on operating in asynchronous mode, refer to *SRDF/Asynchronous Operations* on page 3-7.

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. 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 Version 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 2-64.

Requirements

Dynamic SRDF pairing requires an SRDF licence with Symmetrix Enginuity Version 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 Array Controls CLI Product Guide*.

Note: For Enginuity Version 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 Version 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 2-62.

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 *The Remote Option for Restore, Update, Failback* on page 3-19.

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 Array 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, SyncInProg, Restored, RestoreInProg, SplitInProg.
- The device is the source or target of a TimeFinder/Snap operation.
- There is a background BCV split operation in progress.
- Devices are in the backend Not Ready state.
- There is an optimizer swap in progress on a device.
- The emulation type is not the same (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 Version 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 Version 5671, SRDF/Asynchronous devices are supported for dynamic concurrent RDF.

For information explaining a concurrent SRDF configuration, refer to *Concurrent RDF Operations* on page 3-15.

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.
- 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, SyncInProg, Restored, RestoreInProg, SplitInProg, Updated, UpdateInProg.
- 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.
- 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

Delete One-half of an SRDF Pair

The `half_deletepair` command allows you to dynamically remove the RDF pairing relationship between R1/R2 device pairs. One-half of the specified device pair is converted from an RDF device to a regular device. The command can be specified using a device file or device group. When specified using a device file, all devices listed in the first column of the file will be converted to regular devices (non-RDF). This functionality requires Enginuity Version 5671 or higher.

Note: This command is only allowed for local devices that are Not Ready on the link. Devices cannot be enabled for RDF consistency protection.

For example, to remove the SRDF pairing from device group `Prod` and convert one-half of the paired devices in the group to regular (non-RDF) devices, enter:

```
symrdf -g Prod half_deletepair
```

To remove the SRDF pairing of RDF group 4 on Symmetrix 1123 and convert one-half of those device pairs to regular (non RDF) devices, enter:

```
symrdf half_deletepair -sid 123 -file devicefile -rdfg 4
```

You can use the `symrdf list -half_pair` command to list all half pair devices for a specified Symmetrix or RDF group. Existing half pairs could also be the result of a previous `symstar failover` operation or a configuration change.

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

- Suspended
- Split
- Failed Over
- Partitioned

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

- The device is in one of the following BCV pair states: Synchronized, SyncInProg, Restored, RestoreInProg, SplitInProg, Updated, UpdateInProg.
- 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 are existing local invalid tracks on the local device.
- There are remote invalid tracks and the `-force` option was not specified.
- RDF Consistency is enabled.
- The links are not suspended.

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 -rdfg 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 Version 5x66 or 5267.
- An SRDF license and a Configuration Manager license with Enginuity Version 5567 or higher, and the devices to be swapped are not dynamic RDF-enabled in your Symmetrix configuration.
- An SRDF license with Enginuity Version 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 Version 5567 and 5568 systems are mixed across your local and remote Symmetrix enterprise.
- For Enginuity Version 5669 and higher systems where the R2 device is larger than the R1 device.
- For concurrent RDF devices.

Since Enginuity Version 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 2-14.

Table 2-14 RDF Device Data Status for a Swap

RDF Side With Data	Swap Operation, Refresh Target Selection
RDF1	`-refresh R1` —The R2 device holds the valid copy and the R1 device's invalid tracks will be updated using the R2 data.
RDF1	`-refresh R2` —The R1 device holds the valid copy and the R2 device's invalid tracks will be updated using the R1 data.
RDF2	`-refresh R1` —The R2 device holds the valid copy and the R1 device's invalid tracks will be updated using the R2 data.
RDF2	`-refresh R2` —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 2-15 lists which states are legal for this operation.

Table 2-15 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 Enginuity Version 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 Version 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 2-19 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.

3

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.

RDF Group Topologies in an SRDF

Every RDF device in a Symmetrix array must belong to an RDF (RA) group. There are two types of topologies for RDF 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 3-1, 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.

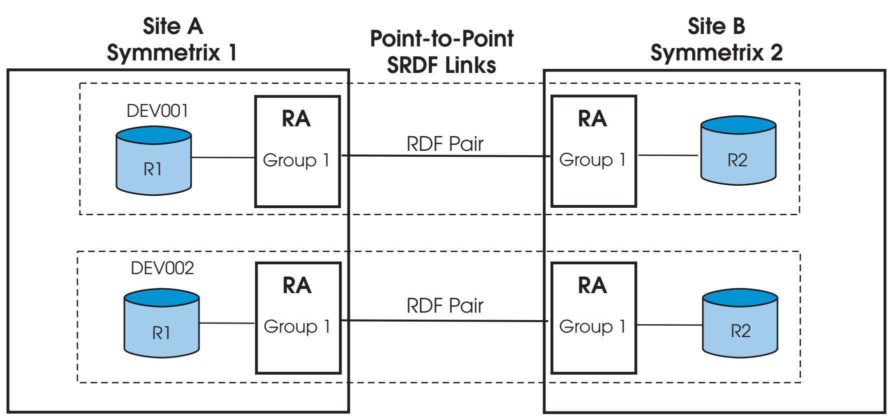

Figure 3-1 RDF Group Topology in a Point-to-Point SRDF Solution

RDF Groups in a Switched SRDF Link

As shown in Figure 3-2, 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.

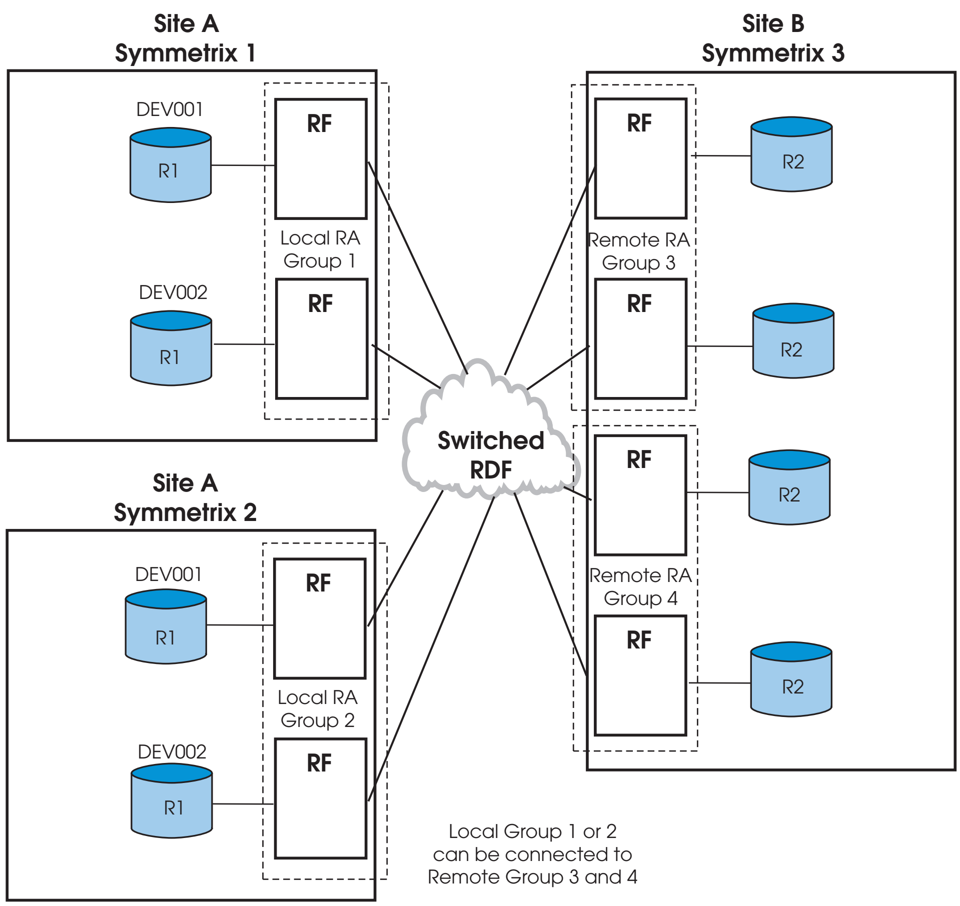

Figure 3-2 RDF Group Topology in a Switched RDF Solution

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
```

Dynamic RDF Group Operations

RDF groups provide a collective data transfer path linking devices of two separate Symmetrix arrays. These communication and transfer paths are used to synchronize data between the R1 and R2 device pairs associated with the RDF 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 3-3 for an example.

RDF groups can be created on demand while the Symmetrix array is in operation. Previously, *static* RDF groups could only be defined at the time of unit configuration. You can add, modify, and delete dynamic RDF groups (RA groups) in a switched fabric SRDF environment.

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 3-2.

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 Version 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 Version 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 *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* for information on how to set Symmetrix metrics.

The following example adds a new dynamic RDF group (`dyngrp4`), 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 (`dyngrp4`) 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 dyngrp4 -rdfg 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 -rdfg 4 -type rdf1
   -invalidate r2
```

For more information on adding dynamic RDF pairs, refer to *Dynamic SRDF Pair Operations* on page 2-61.

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 (`dyngrp4`) to remove a supporting director 13a assigned from the group on the local Symmetrix array 6180.

```
symrdf modifygrp -label dyngrp4 -sid 80 -remove -dir 13a
```

The following example modifies a dynamic group (`dyngrp4`) to add (assign) a supporting director 12a to the group on the local Symmetrix 6180.

```
symrdf modifygrp -label dyngrp4 -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 dyngrp4
```

CAUTION

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

RDF Group Link Limbo

Beginning with Solutions Enabler version 6.1, you can specify the link limbo value for a dynamic RDF group. This advanced user feature allows you to set a specific length of time for Enginuity to wait when a link is detected as down before updating the link status. If the link status is still sensed as Not Ready after the link limbo time expires, devices are then marked Not Ready to the link. The link limbo value that can be set ranges from 0-120 seconds, with the default being 10 seconds. For example, to set the link limbo value to 1 minute for RDF group 4 on Symmetrix array 6180, enter:

```
symrdf -sid 80 -rdfg 4 set link_limbo 60
```

Note: Because the setting of this value affects the application timeout period, it is not recommended to set while running in synchronous mode.

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 2-57.

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 Version 5671. Refer to *Dynamic SRDF Pair Operations* on page 2-61 for additional information on dynamic pairing.
- Concurrent RDF devices that are SRDF/A-capable are supported. Functionality requires Enginuity Version 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 3-15 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 Version 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 3-46 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 Version 5671. Refer to *Mode Transition to Synchronous* on page 3-14.
- 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 Version 5671. Refer to the *EMC Solutions Enabler Symmetrix Array Controls 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 Version 5670 or greater.
- 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[ile] 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, *TimeFinder/Snap and Clone State Reference*.

Note: Refer to *Setting SRDF Modes* on page 2-57 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 Array 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 3-3 for a depiction of SRDF/A delta sets.

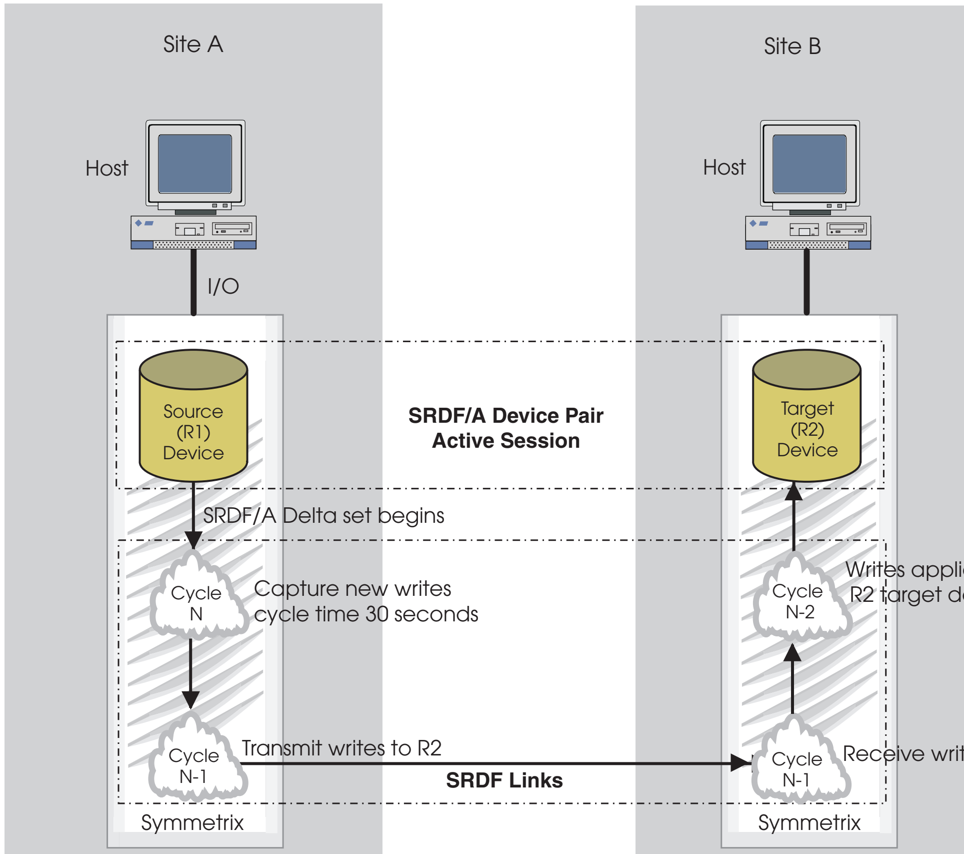

Figure 3-3 SRDF/Asynchronous Mode

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 3-3, 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 *Confirming R2 Data Copy* on page 3-14.

Consistency protection can be enabled and disabled for SRDF/A-capable devices. Refer to *Enabling Consistency Protection with SRDF/A* on page 2-38 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 3-46.

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:

```
RDFA Information:
      {
      Session Number                         10
      Cycle Number                           25
      Number of Devices in the Session       15
      Session Status                         Active

      Session Consistency State              N/A
      Minimum Cycle Time                     00:00:30
      Average Cycle Time                     00:00:30
      Duration of Last Cycle                 00:00:30
      Session Priority                       33

      Tracks not Committed to the R2 Side    1234
      Time that R2 is behind R1              00:00:45
      R1 Side Percent Cache in Use           0
      R2 Side Percent Cache in Use           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 Array 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 Version 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  S T A T E S
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   Suspended
0001 0001    R1:1  ?? RW RW   A..        0         0 RW   WD   Synchronized
0002 0002    R2:2  ?? RW RW   A..        0         0 RW   WD   Synchronized
0003 0003    R2:2  ?? RW RW   A..        0         0 RW   WD   Synchronized
0004 0004    R2:2  ?? RW RW   A..        0         0 RW   WD   Synchronized
```

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          RDF Pair
Device   Dev  E  Tracks    Tracks   S Dev   E  Tracks    Tracks MDAC     STATE
-------------------------------- -- ------------------------ ------ ------------

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
  MB(s)             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 *Synchronizing Changed Tracks* on page 2-27.

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.*

Confirming 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[ile] FileName checkpoint
```

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

```
symrdf -g prod checkpoint
```

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 Version 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.

Consistency Protection

You can enable or disable consistency protection to a device group or file.

For more information, refer to *Enabling Consistency Protection with SRDF/A* on page 2-38.

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 Version 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 3-4, RDF Group 1 is one link to remote Site B and RDF Group 2 is the second link to remote Site C.

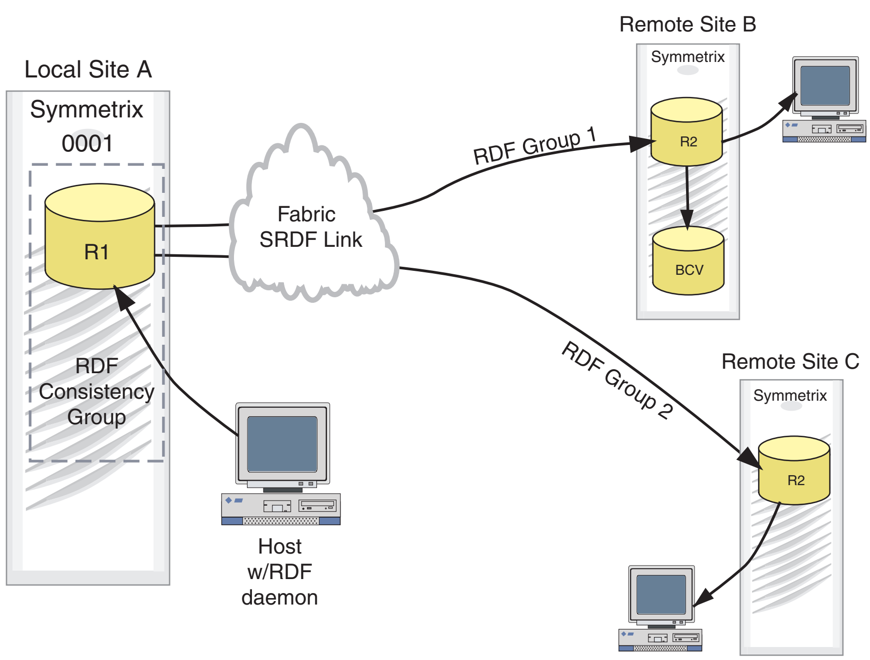

Figure 3-4 Concurrent RDF

Note: Beginning with Enginuity Version 5671, concurrent RDF technology supports the use of composite groups, which can be enabled for consistency protection on both links. Refer to *Composite Group Support* on page 3-17.

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
- Asynchronous

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 Version 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 operations are not supported for concurrent RDF devices:

- Dynamic RDF device operations prior to Enginuity Version 5670
- Beginning with Enginuity Version 5771, semi-synchronous mode is not longer supported.
- PowerPath enabled RDF consistency group operations do not support concurrent RDF devices.

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 3-4), 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.

Composite Group Support

Beginning with Enginuity Version 5671, concurrent RDF devices can be added to a composite group that has been enabled for RDF consistency protection. This allows you to remotely associate a BCV or VDEV across both concurrent R2 mirrors. If both links of the concurrent R1 device are synchronous, you can enable consistency protection on both links at once or on one link. If one link is synchronous and the other is asynchronous, consistency is enabled separately on each link. Both links cannot be asynchronous. Refer to *RDF Consistency Group Operations* on page 3-46 for information on enabling consistency protection for composite groups.

You can create a subset name for a composite group that can be assigned to multiple RDF groups and span multiple source Symmetrix arrays by using the `symrdf -cg set -name` command. The subset composite group must be either all synchronous or all asynchronous. Consistency protection for concurrent RDF can be suspended separately by using the subset name.

For a complete operational example using SYMCLI commands for concurrent RDF using a composite group, refer to *Example 5: Consistency Protection for Concurrent RDF* on page 6-23.

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:

   ```
   symdg 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 3-5, 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.

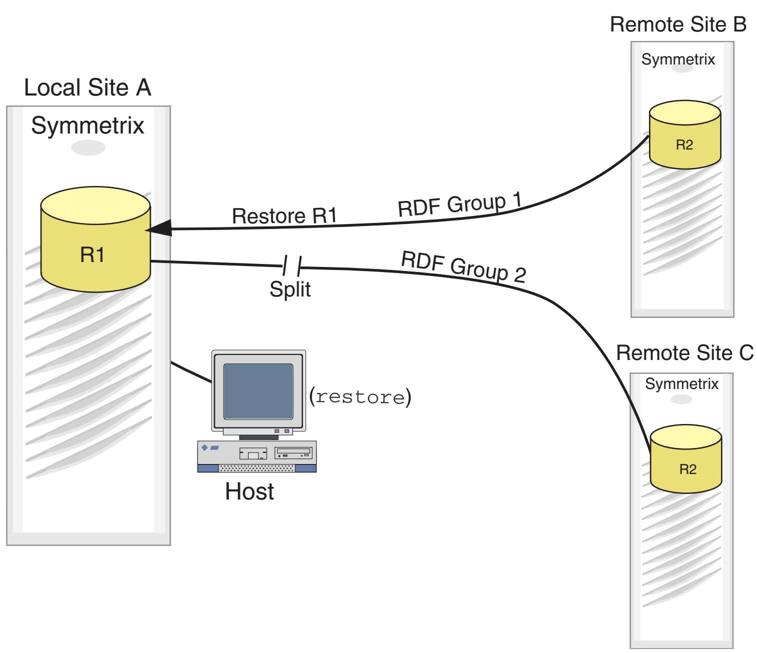

Figure 3-5 Restoring R1 in a Concurrent RDF

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 3-6, once the R1 is restored, then the other R2 mirror is restored.

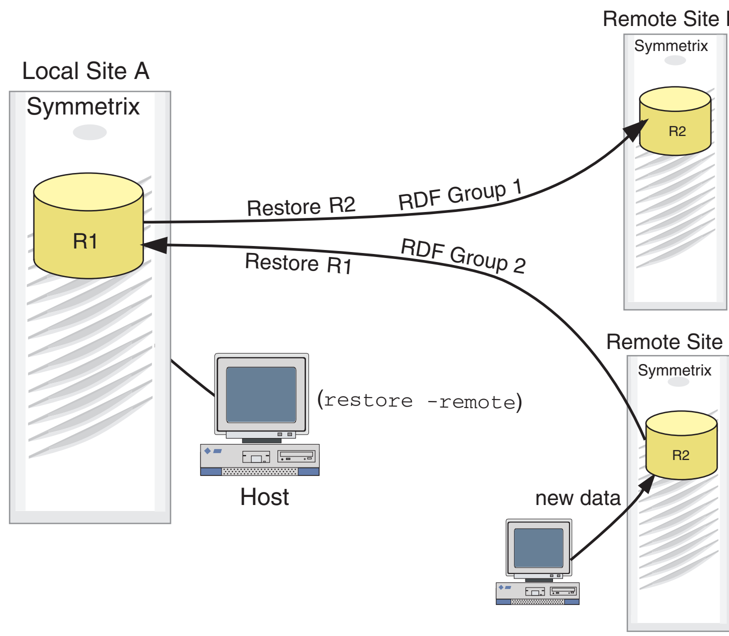

Figure 3-6 Restoring R1 and the Other R2 in a Concurrent RDF

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 Version 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 2-64.

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 3-21.

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 3-7, 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.

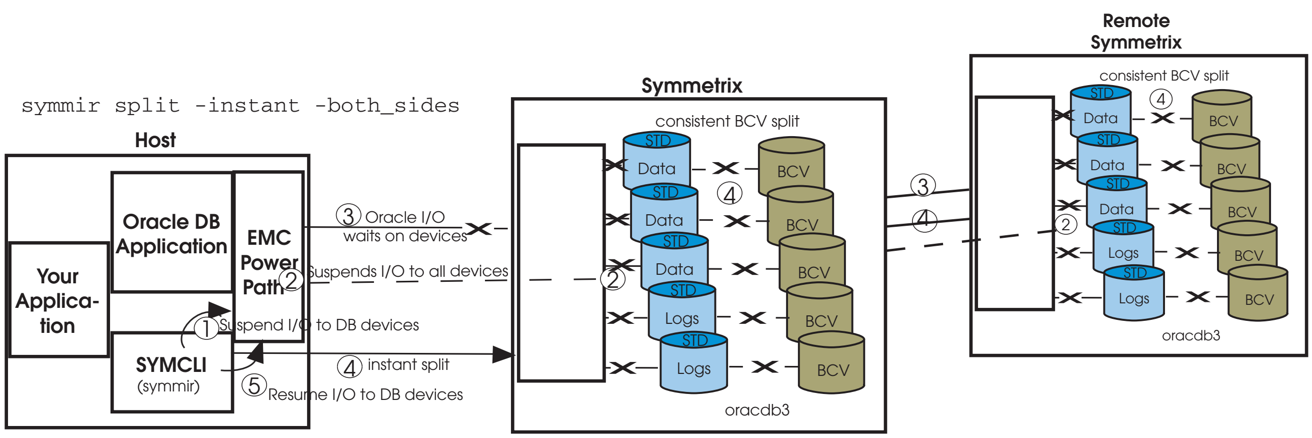

Figure 3-7 Consistent Split on Both Sides

For example in Figure 3-7, the consistent instant split sequence starts with:

`symmir -g oracdb3 split -instant -rdb -dbtype oracle`

1. The `symmir` command sends a suspend I/O log message to PowerPath to suspend I/O on all devices that hold the database.
2. PowerPath suspends I/O to the specified devices where the database devices reside.
3. Oracle cannot write to devices and subsequently waits for devices to become available before resuming any further data I/O.
4. The `symmir` command sends an instants split request to all BCV devices in the specified group, and waits until the split occurs in the device foreground.
5. The `symmir` command sends a resume I/O message to PowerPath.
6. Oracle resumes writing to the devices.

Consistent split actions on both sides of the RDF links can be implemented with an instant (`-instant`) split command, 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]
```

Consistent split actions on both sides of the RDF links can also be implemented with a consistent (`-consistent`) split command, 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 -consistent -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 Version 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 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 3-8 on page 3-22 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.

Note: Beginning with Solutions Enabler 6.1 and Enginuity Version 5671, ECA is also used to provide consistency protection for RDF devices within a composite group that is operating in synchronous mode. Refer to *RDF-ECA Consistency Protection for SRDF/S* on page 3-47.

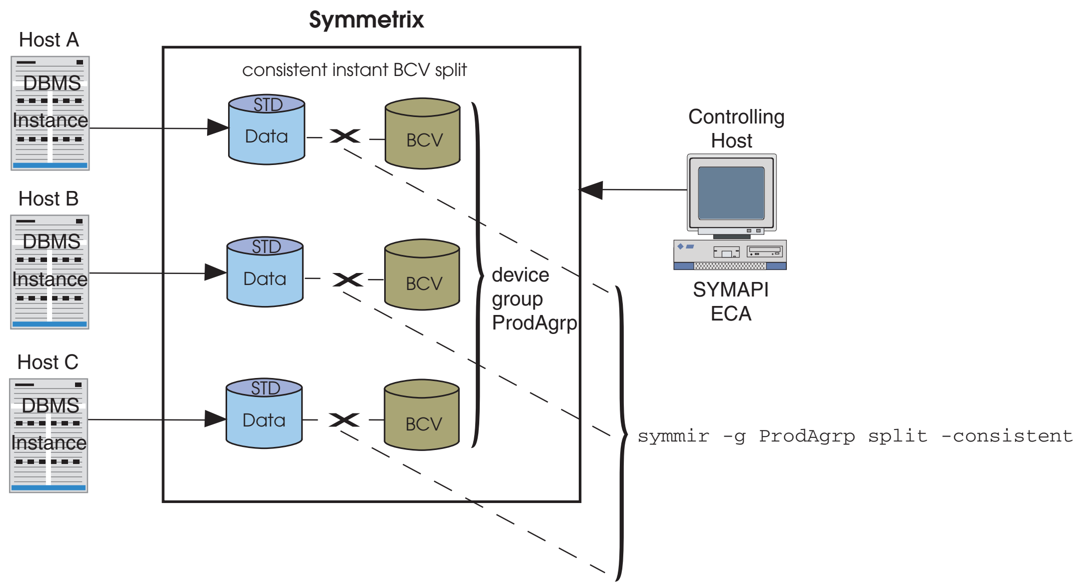

Figure 3-8 ECA Consistent Split

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 3-9 on page 3-25, 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 3-9 on page 3-25.

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*.

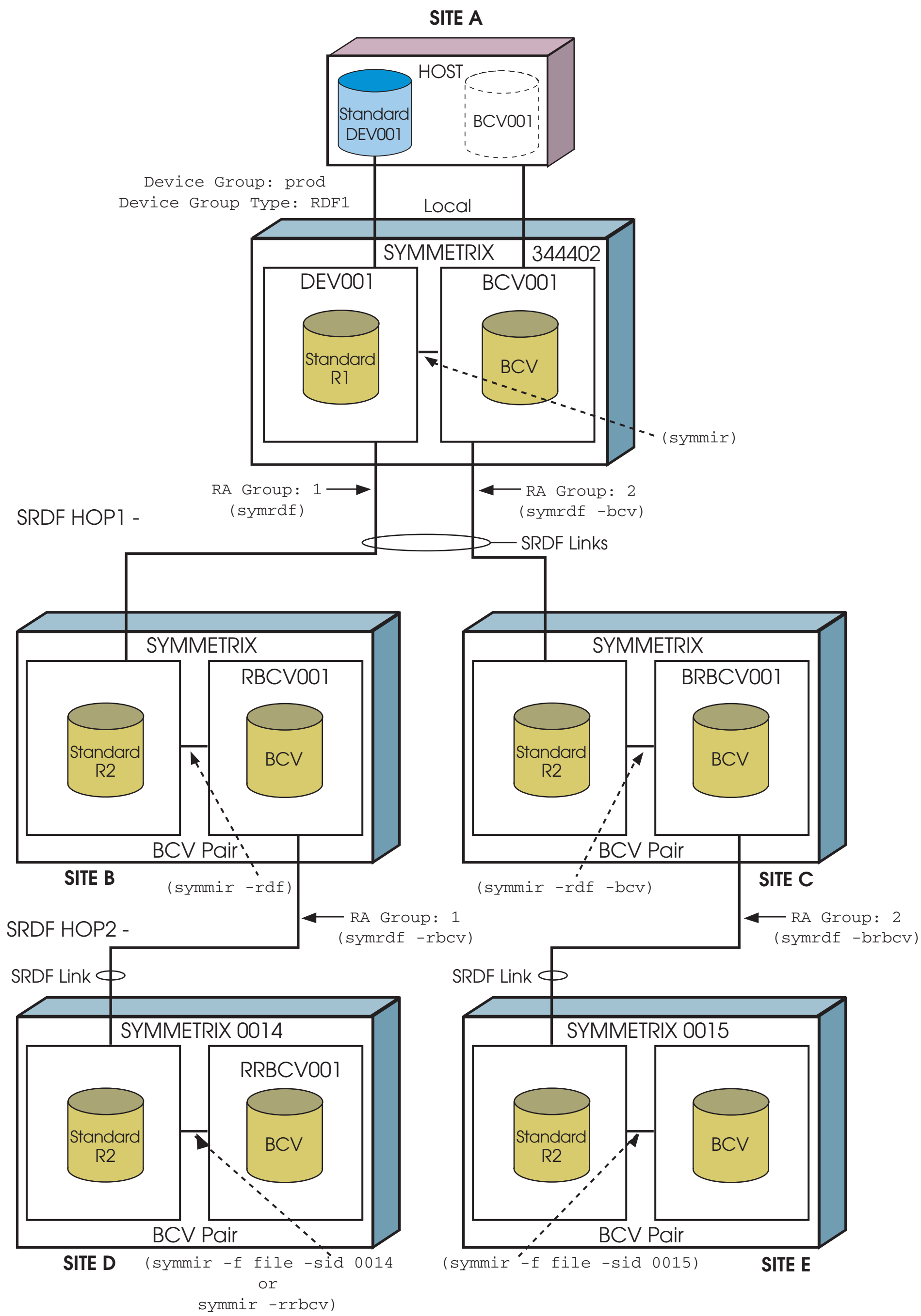

Figure 3-9 Various Remote Configurations

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 3-10 on page 3-27. 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	`symrdf -g <> establish`	Creates the standard associated hop 1 copy.
2	`symmir -g <> split -rdf`	Splits the standard associated hop 1 BCV device pair.
3	`symrdf -g <> establish -rbcv`	Creates the standard associated hop 2 copy.
4	`symrdf -g <> restore -rbcv`	Restores the standard associated hop 1 BCV with the hop 2 copy.
5	`symmir -g <> restore -rdf`	Restores the standard associated hop 1 copy with the hop 1 BCV.
6	`symrdf -g <> restore`	Restores the standard device with the hop 1 copy.
7	`symmir -g <> split`	Splits the standard/BCV pair.
8	`symrdf -g <> establish -bcv`	Creates the BCV associated hop 1 remote copy.
9	`symmir -g <> split -rdf -bcv`	Splits the BCV associated hop 1 device pair.
10	`symrdf -g <> establish -brbcv`	Creates the BCV associated hop 2 copy.
11	`symrdf -g <> restore -brbcv`	Restores the BCV associated hop 1 BCV with the hop 2 copy.
12	`symmir -g <> restore -rdf -bcv`	Restores the standard device associated hop 1 copy with the hop 1 BCV.
13	`symrdf -g <> restore -bcv`	Restores the BCV device with the hop 1 copy.
14	`symmir -g <> restore`	Restores the standard device with the BCV copy.

15	`symmir -f <> -sid 056 establish`
	or
	`symmir -g <> -rrbcv establish`
		Creates the BCV associated hop 2 BCV copy.

16	`symmir -f <> -sid 056 split`
	or
	`symmir -g <> -rrbcv split`
		Splits the BCV-associated hop 2 device pair.

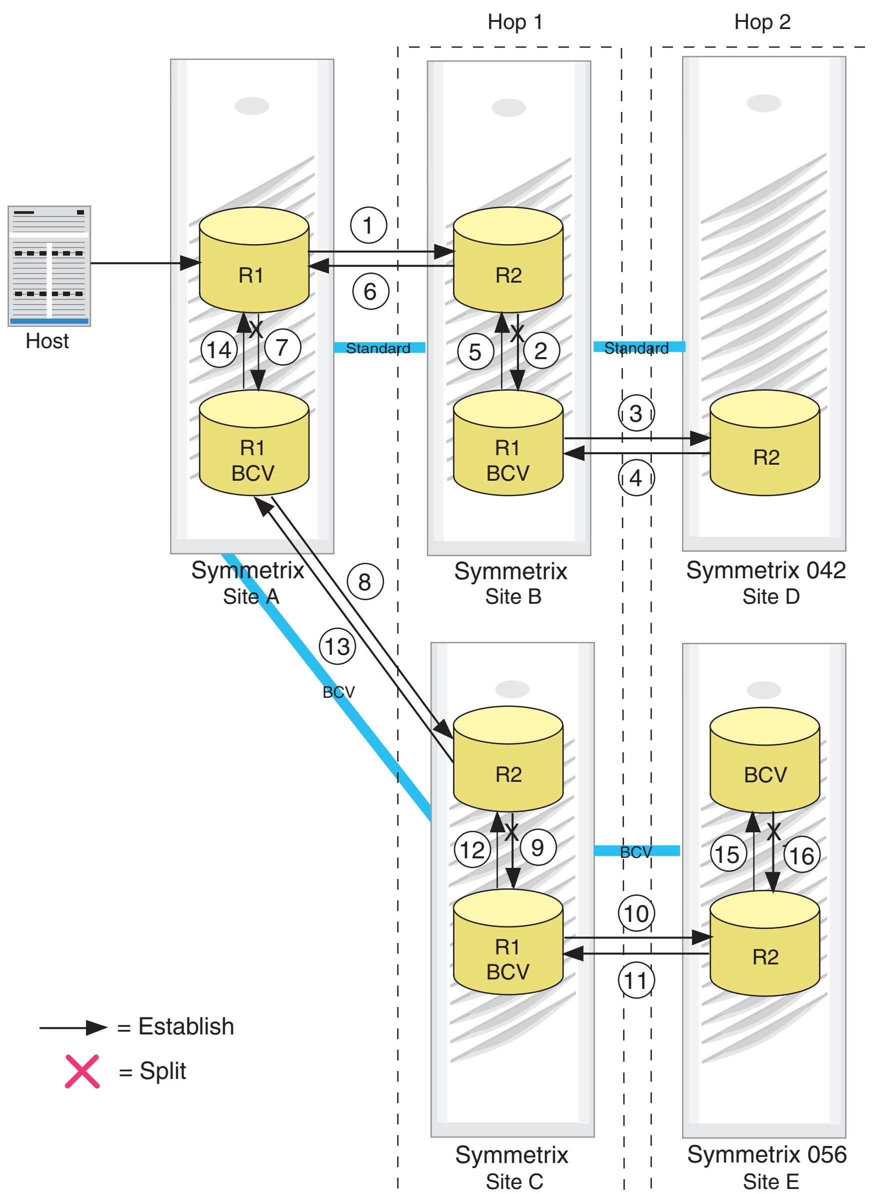

Figure 3-10 Remote Multi-Hop SRDF Configurations

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.

Note: 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 the session executes. For more information, refer to *Locked Devices* on page 3-45.

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).

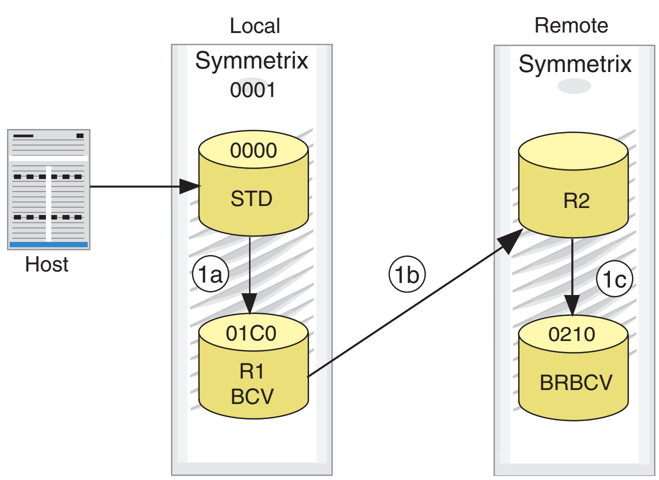

Figure 3-11 Automated Data Copy Path in Single-Hop SRDF Systems

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 3-41) 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.

Note: For expanded operational examples of SRDF/AR in a single-hop configuration, you can refer to Chapter 8, *Performing SRDF/Automated Replication Operations*.

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 3-31. Or, for more information on how to manually set up the multi-hop replication, refer to *Manual Setup for Multi-Hop* on page 3-34.

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 3-41.

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 adaptors), 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.

Beginning with Solutions Enabler 6.1, consistent split operations are automatically retried if the split fails to complete within the allotted timing window. If a consistent split operation fails due to the consistency timing window closing before the split can complete (e.g., `SYMAPI_C_CONSISTENCY_WINDOW_CLOSED`), then the first-hop local BCV device pairs will automatically be resynchronized and the split operation will be reattempted. The consistent split error recovery operation will be attempted the number of times specified in the `SYMCLI_REPLICATE_CONS_SPLIT_RETRY` file parameter, which is defined in the replicate options file. If a value is not specified, then the recovery operation will be attempted 3 times before terminating 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 3-41.

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 3-32, 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 3-41) 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.

Note: For expanded operational examples of SRDF/AR in a multi-hop configuration, you can refer to Chapter 8, *Performing SRDF/Automated Replication Operations*.

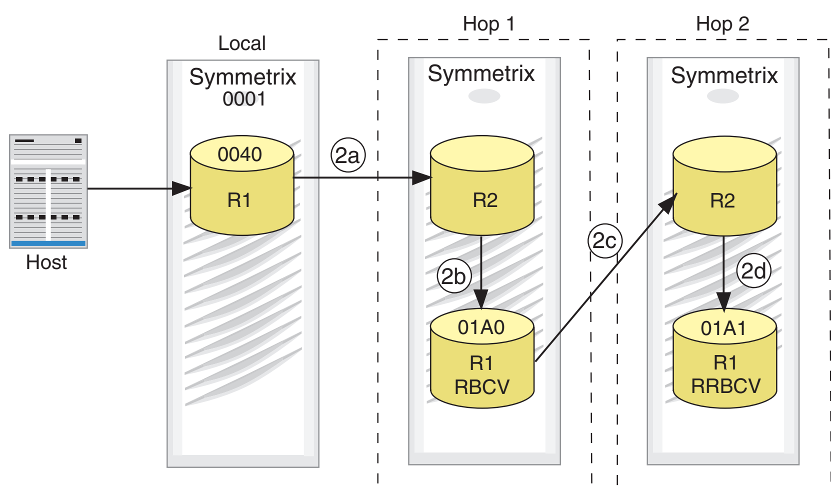

Figure 3-12 Automated Data Copy Path in Multi-Hop SRDF Systems

1. Applies only when you have a final BCV in this Hop 2 Symmetrix path and you have not disabled it.

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 3-29 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.

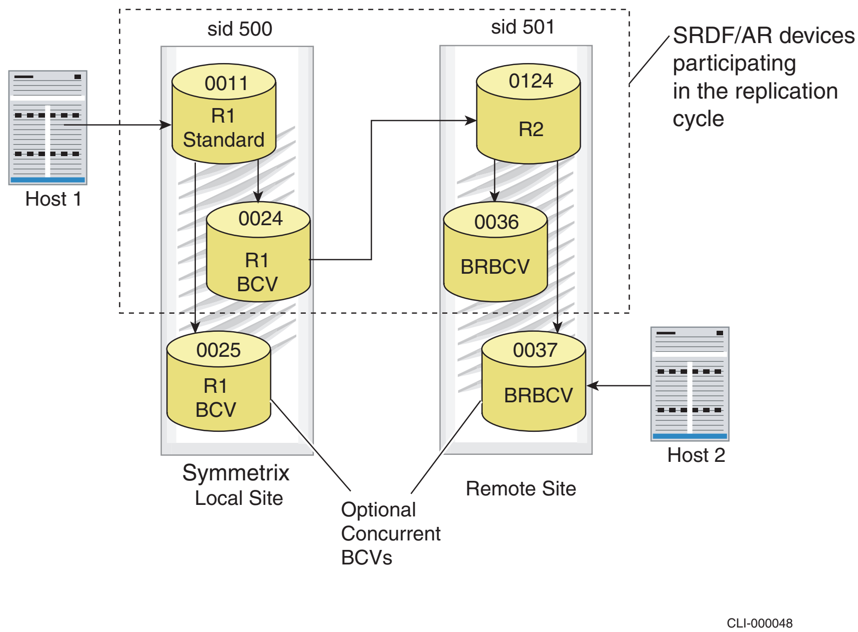

Figure 3-13 Concurrent BCVs in a Single-Hop Configuration

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, 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`.

```
symmir -g srdfar query -multi
```

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.

Note: For an illustrated example, refer to Chapter 8, *Performing SRDF/Automated Replication Operations*.

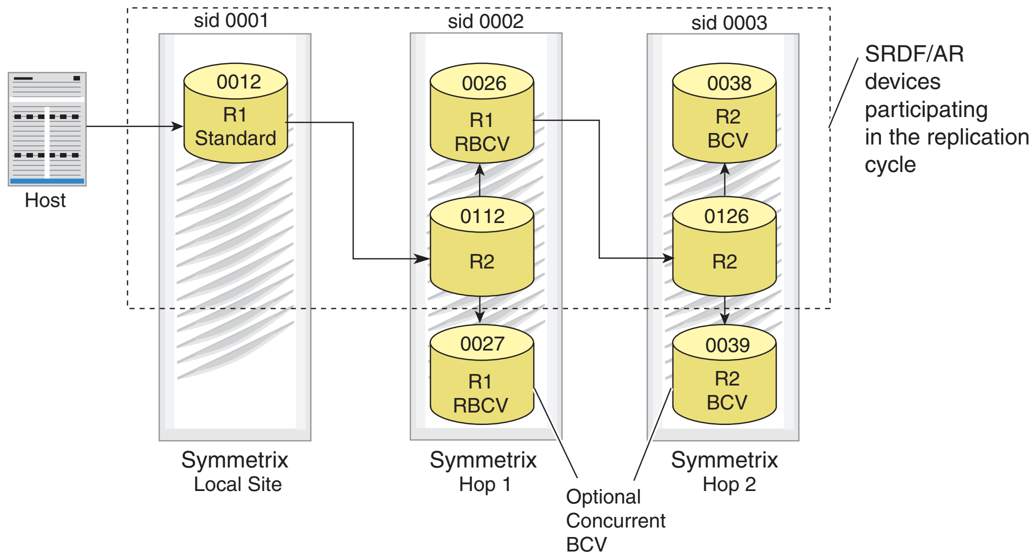

Figure 3-14 Concurrent BCV in a Multi-Hop Configuration

Note: For expanded SRDF/AR operational examples using concurrent BCVs in both the single-hop and multi-hop configurations, you can refer to Chapter 8, *Performing SRDF/Automated Replication Operations*.

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 3-41 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, or every 2, or 3 hours based on data throughput and size.
SYMCLI_REPLICATE_CYCLE=0 SYMCLI_REPLICATE_CYCLE_DELAY=60	Cycle through the first copy, then wait 60 minutes (delay), and then another cycle, delay, 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.

Cycle Time and Invalid Track Statistics

Beginning with Solutions Enabler 6.1, you can display statistical information for cycle time and invalid tracks by using the `symreplicate stats` command. The command can be issued by device group (`-g`) or composite group (`-cg`) for a specified Symmetrix (`-sid`) and information can optionally be written to a specified log file (`-log`). Cycle time (`-cycle`) statistics will be displayed for the last SRDF/AR cycle time, the maximum cycle time and the average cycle time. Invalid track (`-itrks`) statistics will be displayed for the last SRDF/AR cycle, the maximum invalid tracks and the average number of invalid tracks per SRDF/AR cycle. The `-all` option is the default and will display both the cycle time and invalid tracks statistics.

For example, to display both cycle time and invalid track statistics for device group srdfar on Symmetrix 1123, enter:

```
symreplicate -g srdfar -sid 123 -all stats

Group Name: srdfar

Cycle Time (hh.mm.ss):
---------------------------------------

Last Cycle Time:   06:10:01
Max Cycle Time:    08:00:00
Avg Cycle time:    06:00:00

Invalid Tracks:
---------------------------------------

Last Cycle:   12345 ( 9055.5 MB)
Maximum:      10780 ( 8502.3 MB)
Average:      11562 ( 7500.0 MB)
```

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 3-41 for options file syntax.

Clustered SRDF/AR Environments

Since Enginuity Version 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 3-41 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.

SYMCLI_REPLICATE_CONS_SPLIT_RETRY=<*NumRetries*>

Specifies the number of error recovery attempts that will be made when a consistent split operation fails because the timing window closed before the split operation completed. A default retry value of 3 will be used if the SYMCLI_REPLICATE_CONS_SPLIT_RETRY option parameter is not specified when a consistent split (`-consistent`) is requested. A retry value of 0 indicates that no retry attempts should be made.

SYMCLI_REPLICATE_R1_BCV_EST_TYPE=<*EstablishType*>

Specifies the establish type for the local/first hop BCV devices. *EstablishType* specifies the way that BCV establish operations will be executed by TimeFinder. One of the following values may be specified:

`SINGULAR` – BCV devices will be established one at a time; the next device will not be established until the previous device has been established.

`SERIAL` – BCV devices will be established as fast as the establish requests can be accepted by the Symmetrix array.

PARALLEL – BCV devices establish requests will be passed in parallel to each of the servicing DA directors.

SYMCLI_REPLICATE_R1_BCV_DELAY=<*EstablishDelay*>

Denotes how long to wait between issuing establish requests. Establish types of SINGULAR and *PARALLEL*, for an <*EstablishDelay*> can be specified through the SYMCLI_REPLICATE_R1_BCV_DELAY file parameter.

SYMCLI_REPLICATE_FINAL_BCV_EST_TYPE=<*EstablishType*>

Identifies the establish type for the remote/second hop BCV devices.

SYMCLI_REPLICATE_FINAL_BCV_DELAY=<*EstablishDelay*>

Denotes how long to wait between issuing establish requests for the remote/second hop BCV devices. For an establish type of PARALLEL the delay value indicates how long to wait before passing the next establish request to an individual servicing DA director. An establish delay of 0 to 30 seconds may be specified with a value of 0 being the default.

SYMCLI_REPLICATE_ENABLE_STATS=<*TRUE|FALSE*>

Enables or disables the gathering of statistics. By default, statistics gathering is enabled. A value of FALSE indicates that statistics gathering is to be disabled.

SYMCLI_REPLICATE_STATS_RESET_ON_RESTART=<*TRUE|FALSE*>

Resets statistics when a restart action is executed. By default the statistics are not reset upon restart of a symreplicate session. A value of TRUE indicates that statistics are to be reset when restarting a symreplicate session.

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>
SYMCLI_REPLICATE_CONS_SPLIT_RETRY=<NumRetries>
SYMCLI_REPLICATE_R1_BCV_EST_TYPE=<EstablishType>
SYMCLI_REPLICATE_R1_BCV_DELAY=<EstablishDelay>
SYMCLI_REPLICATE_FINAL_BCV_EST_TYPE=<EstablishType>
SYMCLI_REPLICATE_FINAL_BCV_DELAY=<EstablishDelay>
SYMCLI_REPLICATE_ENABLE_STATS=<TRUE|FALSE>
SYMCLI_REPLICATE_STATS_RESET_ON_RESTART=<TRUE|FALSE>
```

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.

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 3-41 for additional information.

RDF Consistency Group Operations

An *RDF consistency group* (SRDF/CG) 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. RDF consistency protection software preserves the dependent-write consistency of devices within the group by monitoring data propagation from source devices to their corresponding target devices. If a source R1 device in the consistency group cannot propagate data to its corresponding R2 device, RDF consistency software suspends data propagation from all the R1 devices in the group. This allows you to quickly recover from certain types of failures or physical disasters by retaining a consistent, DBMS-restartable copy of your database. RDF consistency group protection is available for both synchronous mode (SRDF/S) and asynchronous mode (SRDF/A).

RDF consistency protection for SRDF/S devices is provided using either PowerPath or RDF Enginuity Consistency Assist (RDF-ECA). RDF consistency protection for SRDF/A devices is provided using Multi Session Consistency (MSC).

Note: PowerPath consistency group operations requires Enginuity Version 5265 or higher, PowerPath version 1.5 or higher, and an SRDF/CG license.

Note: RDF consistency group operations for SRDF/A using MSC or RDF-ECA requires Enginuity Version 5671 or higher and an SRDF/CG license.

For information on consistency protection of SRDF/S devices without using PowerPath, refer to *RDF-ECA Consistency Protection for SRDF/S* on page 3-47.

For information on consistency protection of SRDF/A devices, refer to *Multi Session Consistency (MSC) Protection for SRDF/A* on page 3-48.

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 Consistency Group* on page 3-51 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 2-58).

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 3-47), 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 support concurrent RDF devices or SRDF/Star. If devices are configured within an SRDF/Star environment, synchronous consistency protection is provided using RDF-ECA.

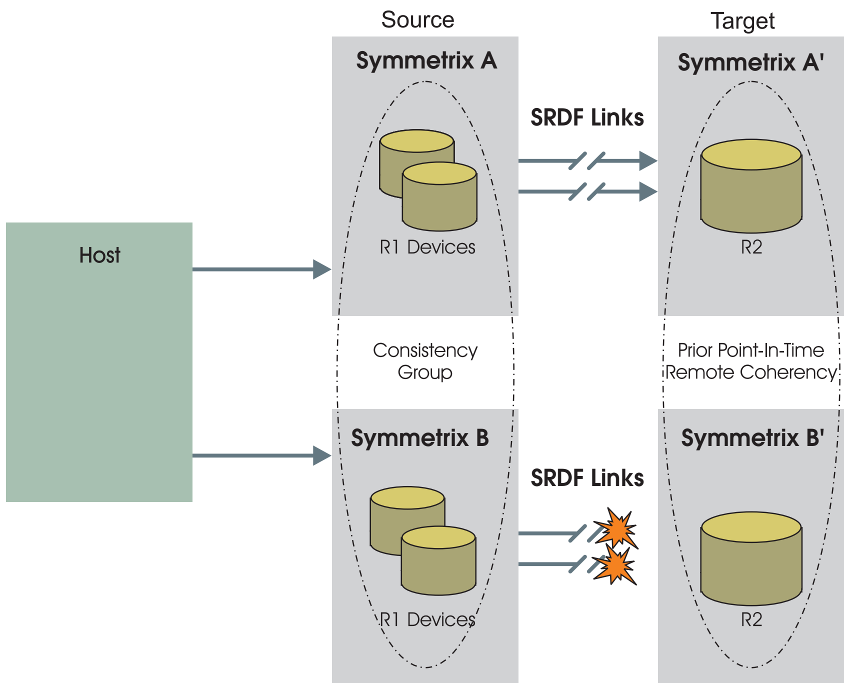

Figure 3-15 All Data Propagation is Suspended Upon Any Link Failure

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 Array Management CLI Product Guide* for additional information on composite groups using the `symcg` command.

Note: For expanded operational examples of PowerPath consistency protection, you can refer to Chapter 7, *Implementing Consistency Protection Using PowerPath*.

RDF-ECA Consistency Protection for SRDF/S

RDF Enginuity Consistency Assist (RDF-ECA) provides consistency protection for synchronous mode devices by performing suspend operations across all SRDF/S devices in a consistency group or a named subset of all devices in a composite group. SRDF/S with RDF-ECA is supported by an RDF daemon that performs monitoring and cache recovery operations across all SRDF/S sessions in the group. If one or more source (R1) devices in an SRDF/S consistency group cannot propagate data to their corresponding target (R2) devices, the RDF daemon suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets.

This ensures that a consistent R2 data copy of the database exists at the point-in-time any interruption occurs. The RDF daemon monitors data copy operations and coordinates the suspension of R1 to R2 data propagation if the consistency protection is suspended (tripped).

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 RDF-ECA consistency group. Refer to *Creating a Consistency Group* on page 3-51 for more information.

Note: For expanded operational examples of RDF-ECA consistency protection, you can refer to Chapter 6, *Implementing Consistency Protection Using RDF-ECA and RDF-MSC*.

Multi Session Consistency (MSC) Protection for SRDF/A

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 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 Consistency Group* on page 3-51 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 expanded operational examples of RDF-MSC consistency protection, you can refer to Chapter 6, *Implementing Consistency Protection Using RDF-ECA and RDF-MSC*.

Using the msc_cleanup Command

The RDF daemon automatically performs MSC cleanup operations for devices in an MSC enabled SRDF/A session during the processing of any RDF control operation. MSC cleanup operations discard any incomplete SRDF/A data or commit completed data to the R2 to maintain dependent write consistency. The MSC cleanup operation can be performed manually by executing the `symrdf msc_cleanup` command for a composite group at the R1 or R2 site, or by RDF group at the R2 site.

For instance, if a failure is detected (e.g., link failure) causing the consistency group to become tripped, the daemon may not be able to process all cleanup operations for the R2 devices where the receive and apply delta sets reside. In this case, the `symrdf msc_cleanup` command can be executed manually from the R2 site. If a consistency group definition is unavailable at the R2 site, the cleanup operation can be performed by directing the command to an RDF (RA) group that was included as part of the consistency group. For example, to perform cleanup operations from the remote host at the R2 site for Symmetrix 1123 and direct the command to RDF group 4, enter:

```
symrdf -sid123 -rdfg 4 msc_cleanup
```

To check whether a MSC cleanup operation is required, use the `symcfg list` command with the `-rdfg all` option to display a list of RDF (RA) groups on a specified Symmetrix array (`-sid`). This command displays flag information for RDF groups operating in SRDF/A mode. The RDFA "Flags M" column denotes whether an MSC cleanup operation is required.

For an example of the `symcfg list` command, refer to *Example 1: Consistency Protection in ASYNC Mode* on page 6-2.

RDF Daemon Support for MSC and ECA

The RDF process daemon maintains consistency for enabled composite groups across multiple Symmetrix arrays for SRDF/A with RDF-MSC and SRDF/S with RDF-ECA. For the consistency 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 (`storrdfd`). 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. RDF consistency 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.

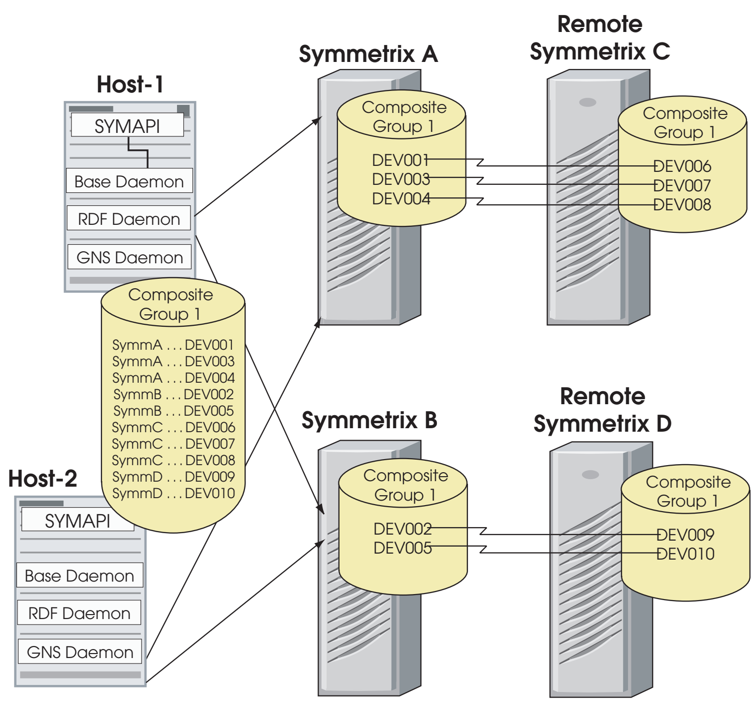

Figure 3-16 RDF Daemon Host Visibility

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 or SRDF/S ECA.

Note: For redundancy, it is recommended that you run multiple instances of the RDF daemon on different hosts. It is also recommended that you do not run the RDF daemon on the host running the applications.

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 and GNS is *not* being used, the daemon will be started automatically by the Solutions Enabler libraries the first time they attempt to connect with it, This may cause a slight delay in performance on that initial connection while the daemon starts and builds its cache.

Note: Prior to starting `storrdfd`, ensure that your default SYMAPI configuration database is up-to-date, since `storrdfd` uses the information stored in it to establish contact with your Symmetrix arrays.

Alternatively, the daemon can be started manually via the `stordaemon` command line utility as follows:

```
stordaemon start storrdfd [-wait Seconds]
```

Note: The `stordaemon` command requires a path of `/usr/storapi/storbin`.

By default, the `stordaemon` 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:

```
stordaemon install storrdfd -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:

```
stordaemon 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 3-3 RDF Daemon Optional Behavior Parameters

Optional Behavior Parameter	= <*OptValue* | *DefaultValue*>	Description
storrdfd:autorestart	= enable | disable	If set to `enable`, 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 Consistency 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 or RDF-ECA consistency protection named `mycg1`, enter:

```
symcg create mycg1 -rdf_consistency -type rdf1
```

The following expanded outline illustrates how to build a consistency group when devices in the group will be either all synchronous or all asynchronous. All devices containing application and system data must be included in the composite group for each DBMS or across the DBMS that controls multi-database transactions.

1. Determine which devices should belong to the consistency group by first listing all RDF (RA) groups on the source Symmetrix arrays connected to the local host. To display devices within a particular RDF group, use the `symrdf list` command with the RDF group number.

   ```
   symcfg list -rdfg all
   symrdf list -rdfg 64
   ```

2. Create a composite group (for example, one named ConsisGrp) on one of the local hosts. Specify the RDF type of the group. Specify the `-rdf_consistency` option to indicate the type of consistency support (PowerPath support is specified with the `-ppath` option).

   ```
   symcg create ConsisGrp -type rdf1 -rdf_consistency
   ```

3. Add to the composite group all devices from one or more RDF (RA) groups. For example, if the RDF groups chosen from the `symcfg list` display are groups number 1 and 64, then all devices in those RDF groups must be managed together during the RDF consistency operation.

   ```
   symcg -cg ConsisGrp -sid 3264 addall dev -rdfg 64
   symcg -cg ConsisGrp -sid 3265 addall dev -rdfg 1
   ```

4. In a database configuration with multiple local hosts, you need to build the same composite group on other local hosts in the configuration. You can use the `symcg export` command to transfer the group definition manually, or GNS to transfer it automatically. The following command creates a text file (`consisgrp.txt`) that contains the composite group definition. You can then use `rcp` (or `ftp`) to transfer that file manually to other local hosts (api28, in this case).

   ```
   symcg export ConsisGrp -f consisgrp.txt
   rcp consisgrp.txt api28:/.
   ```

   Issuing the `symcg import` command on host api28 builds the ConsisGrp composite group on that host, using definitions from the text file. The `-rdf_consistency` option causes the imported group definition to be added to the RDF consistency database on that host.

   ```
   symcg import ConsisGrp -f consisgrp.txt -rdf_consistency
   ```

5. Ensure that all devices in the composite group are either all synchronous or all asynchronous. For example, if the devices are currently operating with synchronous replication and you want them to be operating asynchronously, set the composite group for asynchronous replication:

   ```
   symrdf -cg ConsisGrp set mode async
   ```

6. If the SRDF pairs are not in the Consistent or Synchronized state at this time (for example, the Split or Suspended state with invalid tracks on the R1 side), you can initiate SRDF copying of R1 data to the R2 side. The device state will be SyncInProg until the Consistent or Synchronized state is reached. With asynchronous replication, it may take two cycle switches after all devices reach the Consistent state before the consistency group is consistent.

   ```
   symrdf -cg ConsisGrp establish
   ```

7. From one of the local hosts, enable the composite group for consistency protection (at which time the group becomes a consistency group and is managed by the RDF daemon).

   ```
   symcg -cg ConsisGrp enable
   ```

At this point, each Symmetrix in the configuration watches for any problem with R1 data propagation to the R2 side.

8. If a consistency group configuration includes BCVs at the target site, you can associate these BCVs with the consistency group. Include the RDF group number of the local R1 source devices.

```
symbcv -cg ConsisGrp -sid 3264 associateall dev -range 182:19A -rdf -rdfg 64
symbcv -cg ConsisGrp -sid 3265 associateall dev -range 3B6:3C9 -rdf -rdfg 1
```

9. You can then synchronize the remote BCV pairs. The following command copies data from the R2 devices on the remote Symmetrix arrays to the BCV devices there. The `-rdf` option signifies that the targets are the remote BCVs.

```
symmir -cg ConsisGrp establish -full -rdf
```

Deleting a Consistency Group

You can delete a specified consistency group. If there are members in the group, the command is rejected unless you use the force (`-force`) option.

For example, to delete a consistency group (`mycg1`), regardless if the group has members or not, enter:

```
symcg delete mycg1 -force
```

Note: If the group is enabled for MSC or RDF-ECA consistency protection, the `-symforce` option must be used. The composite group will remain enabled but will be removed from the SYMAPI database.

Note: Deleting the consistency group also stops the RDF daemon from monitoring the composite group.

Refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* for additional information on composite groups using the `symcg` command.

Enabling and Disabling 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 the appropriate SRDF/CG, SRDF/S or 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 either be in asynchronous mode to enable for RDF consistency using MSC or synchronous mode to enable using RDF-ECA.

Note: Domino mode must be disabled to enable the RDF consistency state on the device.

Suspending Consistency Protection

In a consistency group where all devices are either synchronous or asynchronous, you suspend consistency protection for all devices when you issue a `suspend`, `split` or `failover` command to the consistency group. This is sometimes known as manually tripping the group. The difference between `symrdf -cg suspend` and `symrdf -cg split` is the state of the R2 devices at the end of the deactivation.

With `suspend`, the R2 devices are in the write disabled state and cannot be accessed by the target-side hosts, thus maintaining the consistency of the R2 database copy with the production copy on the R1 side.

With `split`, the R2 devices are enabled for both reads and writes by the target-side hosts.

When the same consistency group is defined on multiple hosts, you can initiate a suspend operation from any host provided that the consistency group is enabled. The following command deactivates consistency in a consistency group named ConsisGrp. The `-force` option is required here (and with `split` or `failover`) to ensure that you really want to stop the SRDF mirroring operation and suspend consistency protection.

```
symrdf -cg ConsisGrp suspend -force
```

To resume the RDF links between the SRDF pairs in the consistency group and I/O traffic between the R1 devices and their paired R2 devices, use the `symrdf -cg resume` command:

```
symrdf -cg ConsisGrp resume
```

Consistency protection is automatically restored upon resumption of the link. (Consistency protection is never disabled unless you specifically perform the `symrdf -cg disable` operation.)

For asynchronous replication, you can use the `symrdf -cg verify` command while including the `-cg_consistent` option to ensure that the consistency group is RDF-consistency enabled and in a consistent state. This means that at least two cycle switches have occurred since all devices in each RDF (RA) group reached a consistent state:

```
symrdf -cg ConsisGrp verify -cg_consistent
```

For synchronous replication, verify using the `-synchronized` option:

```
symrdf -cg ConsisGrp verify -synchronized
```

If a consistency group includes BCV devices that are already synchronized with the R2 target devices using symmir establish commands, you deactivate consistency from a local host using a suspend operation and then split all BCV pairs at the target site. The -rdf option tells SYMCLI that you want to split the remote BCV pairs:

```
symrdf -cg ConsisGrp suspend -force
symmir -cg ConsisGrp split -rdf
```

The `symmir query` command displays status information about BCV pairs that are associated with a consistency group.

```
symmir -cg ConsisGrp query -rdf
```

Consistency with a Parallel Database

Figure 3-17 illustrates the use of a consistency group with a parallel database such as Oracle Parallel Server (OPS). The production database system spans two hosts and two Symmetrix arrays (Symmetrix A and C). A user-defined consistency group includes R1 devices from Symmetrix arrays A and C.

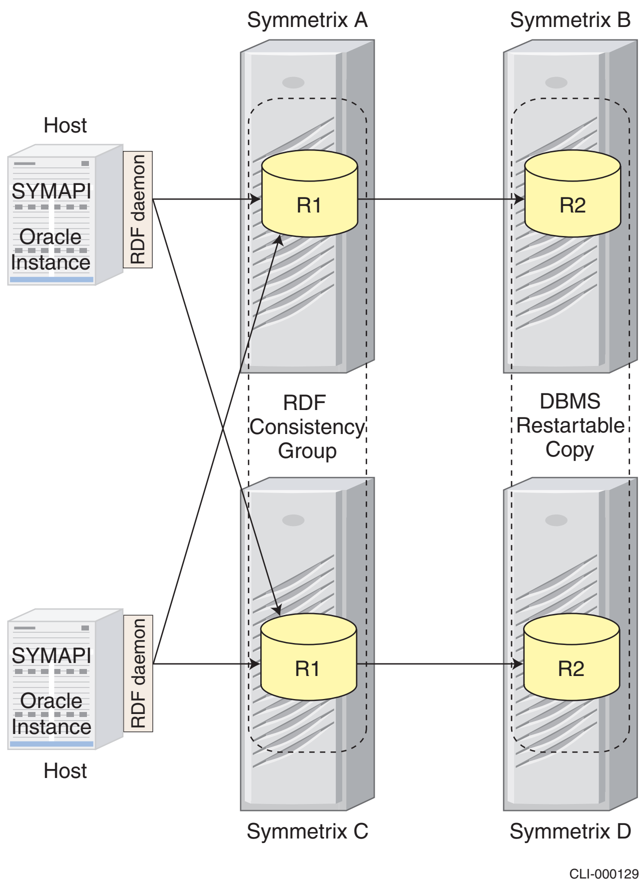

Figure 3-17 Using a Consistency Group with a Parallel Database Configuration

The same consistency group definition must exist on both hosts. If you have enabled Group Name Services (GNS), it automatically propagates a composite group definition to the Symmetrix arrays and to all locally attached hosts that are running the GNS daemon (refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*).

Although each production host can provide I/O to both R1 devices in the configuration, the DBMS has a distributed lock manager that ensures two hosts cannot write data to the same R1 device at the same time. The RDF links to two remote Symmetrix arrays (Symmetrix B and D) enable the R2 devices on those systems to mirror the database activity on their respective R1 devices. A typical remote configuration includes a target-side host or hosts (not shown in the illustration) to restart and access the database copy at the target site.

Although Figure 3-17 shows the RDF daemon located on the production hosts, it is recommended that the RDF daemon be located on control hosts that do not include the production application.

Consistency with BCV Access at the Target Site

When a consistency group includes devices on one or more source Symmetrix arrays propagating production data to one or more target Symmetrix arrays, TimeFinder BCVs at the target site can be indirectly involved in the consistency process.

Figure 3-18 illustrates a configuration with target-side BCVs that mirror the R2 devices. To access data on the BCVs from the target-side host(s), you need to split the BCV pairs at the target sites.

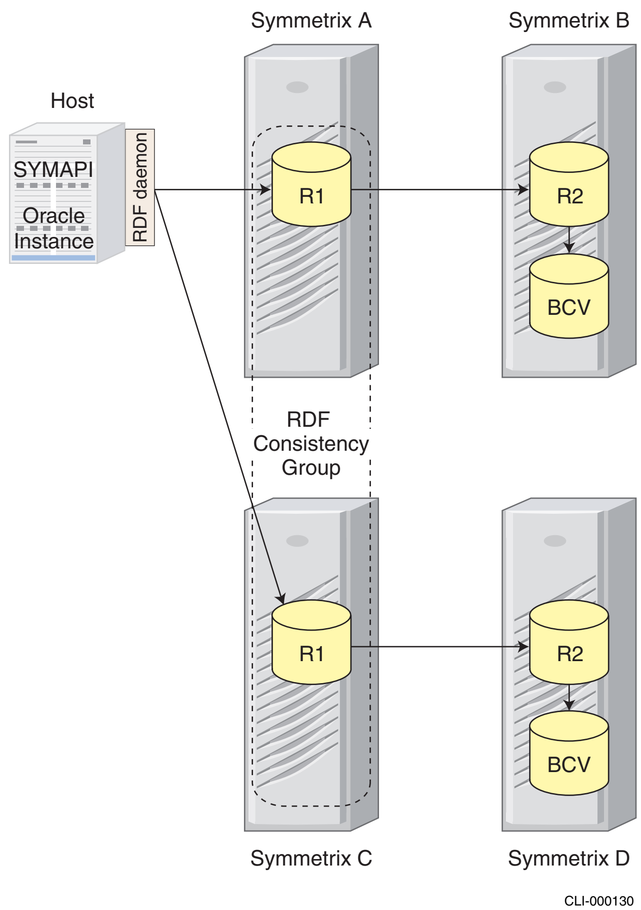

Figure 3-18 Using a Consistency Group with BCVs at the Target Site

The recovery sequence in a configuration that includes BCVs at the target site is the same as described in the previous section except that, at the end of the sequence, the DBMS-restartable copy of the database exists on the target R2 devices and on the BCVs if the BCVs were synchronized with the target site's R2 devices at the time the interruption occurred.

When data propagation is interrupted, the R2 devices of the suspended SRDF pairs are in a Write Disabled state. The target-side host(s) cannot write to the R2 devices, thus protecting the consistent DBMS-restartable copy on the R2 devices. Splitting off the BCV version of the restartable copy allows you to perform disaster testing or business continuance tasks on that data while still maintaining an unchanged R2 copy of the database that can remain consistent with the R1 production database until normal SRDF mirroring between the R1 and R2 sides resumes.

This configuration provides a way to split off and access the DBMS-restartable database copy on the BCVs without risking the data protection that exists on the R2 devices when propagation of data is interrupted.

Managing the BCVs from the R1 side in this configuration depends on the version of Solutions Enabler that you are running. Beginning with Solutions Enabler version 5.4, you can use the consistency group to control a set of SRDF pairs and BCV pairs that spans multiple Symmetrix arrays. In versions of Solutions Enabler prior to version 5.4, you need to control the BCVs using a device group defined for each Symmetrix at the source site.

To manage the BCVs from the R2 side, you can associate the BCVs with a single consistency group defined on a target-site host that is connected to Symmetrix arrays B and D. Or in versions of Solutions Enabler prior to Version 5.4, you can include the BCVs in device groups defined for each of these two Symmetrix arrays.

Although Figure 3-18 shows the RDF daemon located on the production host, it is recommended that the RDF daemon be located on control hosts that do not include the production application.

Creating Composite Groups from Various Sources

If you already have existing sources that define the devices that you want to include in your composite group, you can translate one of these sources into a new composite group rather than build the composite group explicitly as described previously in *Creating a Consistency Group*. You can also use this method to translate an existing source into an existing composite group.

Creating a Composite Group from an Existing Device Group

You can translate the devices of an existing device group to a new or existing composite group. The following command translates and adds all devices from a device group named Symm64DevGrp to a composite group named ConsisGrp. The `-rdf_consistency` option adds the composite group to the RDF consistency database on the host and makes the group capable of being enabled for RDF consistency protection.

```
symdg dg2cg Symm64DevGrp ConsisGrp -rdf_consistency
```

Creating a Composite Group from an RDBMS Database

You can also translate the devices of an existing RDBMS database or tablespace to a new or existing composite group. However, for SYMCLI to access a specified database, you need to first set the `SYMCLI_RDB_CONNECT` environment variable to the username and password of the system administrator's account. For example, when connecting locally, you can use the following command to set the variable to a username of "system" and a password of "manager." (The Bourne and Korn shells use the `export` command to set environment variables; the C shell uses the `setenv` command).

```
export SYMCLI_RDB_CONNECT=system/manager
```

When connecting via the network, you need to add a database-specific variable to the `RDB_CONNECT` definition. For example, connecting via the network in an Oracle environment means that you have an Oracle network listener process running. In this case, you need to add an Oracle connection string such as the Transparent Network Substrate (TNS) alias name "api217" in the following command.

```
export SYMCLI_RDB_CONNECT=system/manager@api217
```

Similarly, connecting via the network in an SQL Server 2000 environment requires adding a string such as "HR" to indicate the ODBC data source administrator.

```
set SYMCLI_RDB_CONNECT=system/manager@HR
```

Optionally, you can set the SYMCLI_RDB_TYPE environmental variable to a specific type of database (for example, oracle, informix, sqlserver, or ibmudb) so that you do not have to include the -type option on the symrdb rdb2cg command line. The following command sets this variable to oracle.

```
export SYMCLI_RDB_TYPE=oracle
```

The following symrdb rdb2cg command translates the devices of an oracle type database named oradb to an RDF1 type composite group named ConsisGrpDb. The -rdf_consistency option adds the composite group to the RDF consistency database on the host.

```
symrdb -type oracle -db oradb rdb2cg ConsisGrpDb -cgtype rdf1
   -rdf_consistency
```

The following symrdb tbs2cg command translate the devices of an oracle type tablespace named orats to an RDF1 type composite group named ConsisGrpTs.

```
symrdb -type oracle -tbs orats tbs2cg ConsisGrpTs -cgtype rdf1
   -rdf_consistency
```

For a list of currently supported databases and platforms, refer to *EMC Solutions Enabler Support Matrix*.

With most RDBMS database systems, it is necessary to set up environment variables that are specific to that system. For example, Oracle systems use ORACLE_HOME and ORACLE_SID, and Sybase systems use SYBASE and DSQUERY. *Example 4: Creating a Composite Group from Existing Sources* on page 6-19 shows how to define the Oracle environment variables.

Creating a Composite Group from a Logical Volume Group

You can also translate the devices of an existing logical volume group to a new or existing composite group using the symvg command. This command does not require setting up any environment variables before performing this operation.

To translate the devices of a logical volume group named LVM4vg to an RDF1 type composite group named ConsisGrp. The -rdf_consistency option adds the composite group to the RDF consistency database on the host.

```
symvg vg2cg LVM4vg ConsisGrp -cgtype rdf1 -rdf_consistency
```

For a list of currently supported Logical Volume Managers and platforms, refer to *EMC Solutions Enabler Support Matrix*.

4

SRDF/Star

This chapter discusses SRDF/Star and the use of the SYMCLI `symstar` command. It explains how to setup, manage and failover data operations in the event of a site or link failure.

Note: To perform SRDF/Star operations with Access Control Enabler, you need RDF BASECTRL, BASE, and BCV access types. For more information, refer to *EMC Solutions Enabler Access Control CLI Product Guide*.

Introduction to SRDF/Star

SRDF/Star is a three-site[1] disaster recovery solution that uses concurrent RDF technology to replicate data from a primary production site (referred to as the *workload site*) to both a nearby remote site and a distant remote site. Data is transferred in SRDF/Synchronous (SRDF/S) mode to the nearby remote site (referred to as the *synchronous target site*) and in SRDF/Asynchronous (SRDF/A) mode to the distant remote site (referred to as the *asynchronous target site*). Refer to Figure 4-1, *SRDF/Star Configuration* for a depiction.

SRDF/Star provides consistent data protection and incremental data recovery between target sites in the event of a workload site failure or transient fault (e.g., link failure). In the event that the workload site becomes inoperable, SRDF/Star provides failover capability through incremental recovery to quickly re-establish data replication operations between the target sites, via SRDF/A recovery links. Either of the two target sites could resume data operations for the workload site, while the other would resume as a protected secondary target site. SRDF/A recovery links and additional control hosts are required at the target sites to continue data processing operations in the event of a failure.

Data is protected through enhanced RDF-MSC and RDF-ECA consistency group technology, which monitors the data propagation from the source volumes to their corresponding target volumes. Devices within the enabled consistency group are protected to preserve the dependent-write consistency of a database, which may be distributed across multiple RDF platforms. For more information on the consistency group technology employed for SRDF/Star, refer to *RDF Consistency Group Operations* on page 3-46.

Note: The SRDF/Star requires Enginuity Version 5671 or higher and an SRDF/Star license.

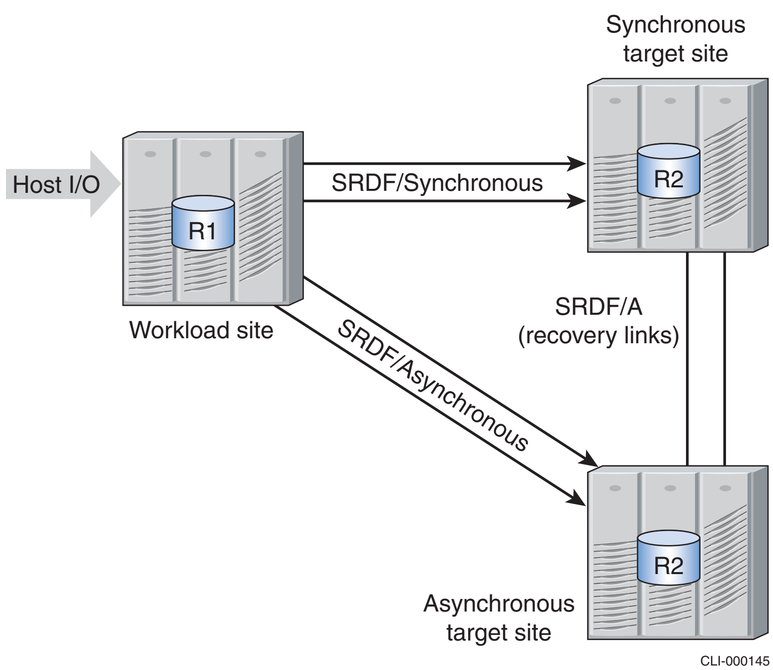

Figure 4-1 SRDF/Star Configuration

1. SRDF/Star is implemented as a triangular topology, which may be expanded to include multiple Symmetrix triangles.

SRDF/Star Benefits and Features

SRDF/Star provides advanced multi-site business continuity protection. It combines RDF technologies to enable concurrent SRDF/S and SRDF/A operations from the same source volumes at the workload site with the ability to incrementally establish an SRDF/A session between the two target sites in the event of a workload site outage — a capability only available through SRDF/Star software. SRDF/Star is a combination of host software and Enginuity functionality that operates in a concurrent SRDF configuration, where one remote mirror operates in SRDF/S mode and the other in SRDF/A mode.

SRDF/Star provides a rapid re-establishment of replication operations in the event of workload site failure. Rather than performing a full resynchronization between the asynchronous and synchronous target sites, SRDF/Star performs a differential synchronization, which dramatically reduces the time it takes to remotely protect the new workload site after the primary site failure. SRDF/Star also provides a mechanism for the user to determine which site contains the most current data in the event of a rolling disaster that affects the workload site. In all cases, the choice of which site to use in the event of a failure is left to the discretion of the customer.

The Solutions Enabler SRDF/Star provides the following benefits and features:

- Sites can be geographically dispersed.
- SRDF/Star can span multiple RDF groups and Symmetrix arrays.
- SRDF/Consistency Groups (SRDF/CG) maintain data consistency across SRDF/Star.
- In the event of a workload site failure, SRDF/Star enables you to failover and resume asynchronous data transfer between the remaining target sites.
- Data is synchronized differentially, so the time to establish remote mirroring and consistency is minimal.
- In the event of a rolling disaster affecting the workload site, you can determine which of the target sites (synchronous or asynchronous) holds the more current data and switch operations to that site.

Device Restrictions

Solutions Enabler SRDF/Star has the following device restrictions:

- CKD-striped meta devices are not supported.
- SRDF/Star does not support R2 devices that are larger than the R1 device.
- BCV device management must be configured separately.

> Note: It is strongly suggested that you have BCV device management available at both the synchronous and asynchronous target sites. Refer to *Step 6: Adding BCV Devices to the SRDF/Star Configuration* on page 4-10.

- RDF groups cannot be shared between separate SRDF/Star configurations.
- The R2 devices must not be BCV devices.

SRDF/Star Failure Scenarios

SDRF/Star is fault tolerant and highly available when constituent components such as device RAID mirroring, redundant RDF directors, and redundant RDF links between sites are configured. However, within SRDF/Star, the inter-site paths (workload-to-synchronous, workload-to-asynchronous, and synchronous-to-asynchronous) are independent of one another. This means that you need to consider the point and the time of faults that lead to failures to determine whether a recovery scenario exists.

Single component faults that leave the system in a degraded but operational state do not require SRDF/Star recovery actions. However, if component faults persist and additional component faults occur, an SRDF/Star fault may be created.

Within SDRF/Star, faults are considered relative to where the production workload is running (the workload site). Therefore, the definition of an SRDF/Star fault is any combination of component faults that persist and cause either of the following conditions:

- The triggering and tripping of consistency protection, which halts all I/O flow from the workload site to the remote site and leaves the remote site data image in a consistent state.
- Loss of a workload site, which leaves remote site data images in a consistent state.

Transient faults, coupled with the fact that multiple faults may occur within SRDF/Star at any point and in any sequence, create a number of transient and disaster fault scenarios:

- Single Transient Faults

  Single network or remote site faults with recovery procedures that do not disrupt the workload site.

- Multiple Transient Faults

  Multiple network or remote site faults with recovery procedures that do not disrupt the workload site but which require recovery procedure serialization to guarantee that SRDF/Star does not completely expose the workload site without any remote protection.

- Disaster Faults

  Single and/or multiple transient faults followed by a workload site loss that requires moving the workload-site to one of the remote sites within SRDF/Star.

Setting up SRDF/Star

Before you can make SRDF/Star operational, you must perform the following tasks:

1. Verify the settings for each Symmetrix array to be included in the SRDF/Star configuration. For instructions, refer to *Step 1: Verifying Symmetrix Settings* below.
2. Create a composite group at the workload site. For instructions, refer to *Step 2: Creating SRDF/Star Composite Groups* on page 4-6.
3. Create an SRDF/Star options file containing specific parameters for the automated setup procedure. For instructions, refer to *Step 3: Creating the SRDF/Star Options File* on page 4-8.
4. Use the `symstar setup` command to build the internal SRDF/Star definition file and copy this file to other control hosts. For instructions, refer to *Step 4: Performing the symstar setup Operation* on page 4-8.
5. Optionally, use the `symstar buildcg` command to build matching R2 composite groups at the target sites. For instructions, refer to *Step 5: Creating R2 Composite Groups* on page 4-9.
6. Optionally, add BCVs to the SRDF/Star configuration. For instructions, refer to *Step 6: Adding BCV Devices to the SRDF/Star Configuration* on page 4-10.

The examples in this section use the name `StarGrp` for the composite group and the names `NewYork`, `NewJersey`, and `London` as the names for the workload site, the synchronous target site, and the asynchronous target site, respectively.

Note: To see expanded operational examples for SRDF/Star, you can refer to the following Technical Note, *Using SYMCLI to Implement SRDF/Star.*

Step 1: Verifying Symmetrix Settings

Verify the following requirements:

❑ Each Symmetrix array within SRDF/Star uses dynamic RDF devices (capable of being dynamically converted to either an R1 or an R2 device).

❑ The RDF directors are Fibre or Gig-E (RF or RE).

❑ The following states exist for each Symmetrix array within SRDF/Star (use the `symcfg list -v` command to display a Symmetrix configuration):

- Concurrent RDF Configuration State = Enabled
- Dynamic RDF Configuration State = Enabled
- Concurrent Dynamic RDF Configuration = Enabled
- RDF Data Mobility Configuration State = Disabled

❑ Each RDF group in the composite group has the following configuration (use the `symcfg list -rdfg -v` command to display):

- Prevent RAs Online Upon Power On = Enabled
- Prevent Auto Link Recovery = Enabled

Note: Preventing automatic recovery preserves the remote copy that was consistent at the time of the link failure.

Step 2: Creating SRDF/Star Composite Groups

An RDF1 type composite group must be created on the control host for the Symmetrix array at the workload site. When created, the composite group is enabled for consistency protection (thus, referred to as a consistency group). The consistency group is then populated with devices from the concurrent RDF groups. Refer to Figure 4-2.

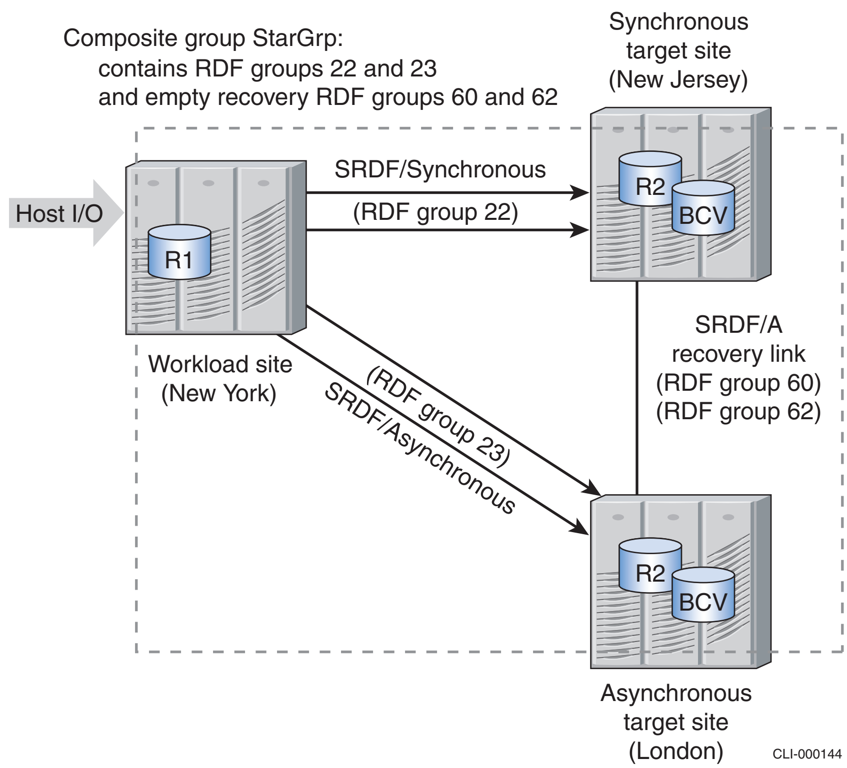

Figure 4-2 Star Composite Group

The following procedure explains how to build an RDF1 type composite group on the control host of the SRDF/Star workload site (that is, NewYork, Symmetrix 11). The R1 devices must be configured as concurrent dynamic devices (dynamic devices that are capable of being either an R1 or an R2 device). For illustrative example, the synchronous target site is in New Jersey and the asynchronous target site is in London.

1. Determine which devices on the local Symmetrix array (`-sid 11`) are configured as concurrent dynamic devices:

```
symrdf list -sid 11 -concurrent -dynamic -both
```

Note: Use the `-dynamic` and `-both` options to display dynamic SRDF pairs in which the paired devices can be either R1 or R2 devices.

2. Create an RDF1-type composite group (for example, one named StarGrp) on the control host at the workload site.

   ```
   symcg create StarGrp -type rdf1 -rdf_consistency
   ```

   Note: Use the `-rdf_consistency` option to specify consistency protection for the group.

3. Add devices to the composite group from those RDF groups that represent the concurrent links for the SRDF/Star configuration:

   ```
   symcg -cg StarGrp -sid 11 addall dev -rdfg 23
   ```

   Note: With concurrent RDF, the command that adds one of two concurrent groups (for example, RDF group 23) actually adds both concurrent groups (for example, synchronous RDF group 22 and asynchronous RDF group 23).

4. Create two RDF group names - one RDF group name for all synchronous links and one for all asynchronous links:

   ```
   symcg -cg StarGrp set -name NewJersey -rdfg 11:22
   symcg -cg StarGrp set -name London -rdfg 11:23
   ```

   Note: The name `NewJersey` includes synchronous RDF group 22 on Symmetrix 11 (although you could include other synchronous RDF groups by using the `sid:rdfg` syntax). The name `London` includes concurrent asynchronous RDF group 23.

   You must also include the names `NewJersey` and `London` in an SRDF/Star options file as the values for the synchronous and asynchronous target site names, respectively. For more information, refer to *Step 3: Creating the SRDF/Star Options File* on page 4-8.

5. For each source RDF group that you added to the composite group, define a corresponding recovery RDF group at the remote site. A recovery RDF group can be static or dynamic, but it cannot be shared. A recovery RDF group must also be empty (that is, it cannot contain any devices). For example, RDF group 60 is an empty static or dynamic group that has been configured on the remote Symmetrix array to which source RDF group 22 is linked. Recovery RDF group 62 has been configured on the other remote Symmetrix array as a match for source RDF group 23. To add a corresponding recovery RDF group at the remote sites, enter:

   ```
   symcg -cg StarGrp set -rdfg 11:22 -recovery_rdfg 60
   symcg -cg StarGrp set -rdfg 11:23 -recovery_rdfg 62
   ```

   Note: These two recovery group definitions represent one recovery RDF group as viewed from each of the two target sites. These two definitions for the unique pairing that is necessary for recovery operations.

Note: Refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* for additional information on composite groups and using the `-cg` command.

Step 3: Creating the SRDF/Star Options File

An SRDF/Star options file (a text file) specifying the names of each SRDF/Star site and other required parameters must be created. The options file can contain comment lines that begin with the pound sign (#). This file should conform to the following syntax, where you enter a value after the equal sign (=):

```
#Comment
SYMCLI_STAR_WORKLOAD_SITE_NAME=WorkloadSiteName
SYMCLI_STAR_SYNCTARGET_SITE_NAME=SyncSiteName
SYMCLI_STAR_ASYNCTARGET_SITE_NAME=AsyncSiteName
SYMCLI_STAR_ADAPTIVE_COPY_TRACKS=NumberTracks
SYMCLI_STAR_ACTION_TIMEOUT=NumberSeconds
```

The *WorkloadSiteName* value should be a name that is meaningful for the workload site. For example, if this site is located in New York, the site name might be NewYork or New_York. The default is Site_A.

The *SyncSiteName* value should be a name that is meaningful for the synchronous target site. For example, if this site is located in New Jersey, the site name might be NewJersey. This name must match the RDF group name that you use for the synchronous RDF groups when building the composite group. The default is Site_B.

The *AsyncSiteName* value should be a name that is meaningful for the asynchronous target site. For example, if this site is located in London, the site name might be London. This name must match the RDF group name that you use for the asynchronous RDF groups when building the composite group. The default is Site_C.

The *NumberTracks* value is the number of invalid tracks that need accumulate before transitioning from Adaptive Copy mode into the SRDF mode. The default is 30,000.

The *NumberSeconds* value is the maximum time (in seconds) that the system waits for a particular condition before returning a time-out failure. The wait condition may be the time to achieve R2-recoverable SRDF/Star protection or RDF consistency protection, or the time for RDF devices to reach the specified number of invalid tracks while synchronizing. The default is 1800 seconds (30 minutes). The smallest value allowed is 300 seconds (5 minutes).

Step 4: Performing the symstar setup Operation

The SRDF/Star symstar setup command reads and validates the information in the host composite group definition and builds the internal SRDF/Star definition file that defines the R1 composite group for the workload site. This information is combined with the options from your SRDF/Star options file and written in an internal format to the definitions folder of the SYMAPI installation path. To set up for possible recovery operations at the remote sites, you will need to manually copy this internal definition file to control hosts at the synchronous and asynchronous target sites.

Note: If control operations will be performed from multiple control hosts at the workload site, then the internal definition file must also be copied to those hosts.

Once the internal definition file is copied to the remote hosts, you can use the symstar buildcg command to build the R2 composite groups needed for recovery operations at the synchronous or asynchronous target site. An alternative is to build the matching R2 composite groups manually at those sites (as was done above).

To execute the setup operation for a composite group named StarGrp and include option values from an SRDF/Star options file named MyOpFile.txt, enter:

```
symstar -cg StarGrp -options MyOpFile.txt setup
```

The setup procedure builds an internal SRDF/Star definition file that defines the R1 composite group at the workload site. This definition file has the same name as the composite group and is saved in UNIX to `/var/symapi/config/STAR/def/CGname` (for example, `/var/symapi/config/STAR/def/StarGrp`). In Windows, the file is saved to `\Program Files\EMC\SYMAPI\config\Star\def\CGname`. You must then manually copy this file to the same definitions folder on the control hosts at the synchronous and asynchronous target sites.

The `-clear` option can be used with the `symstar setup` command to change the star mode setting of all participating RDF groups to *OFF*.

Note: The `-clear` option should only be used when SRDF/Star is disabled and both target sites are unprotected.

The `-reload_options` parameter can be used with the `symstar setup` command to update the options values in the SRDF/Star definition file. It cannot be used to update any site name values.

Step 5: Creating R2 Composite Groups

Optionally, after copying the definition file, you can then use the `symstar buildcg` command (from any host that has the definition file) to create the matching R2 composite groups needed for recovery operations at the synchronous and asynchronous target sites.

```
symstar -cg StarGrp buildcg -site NewJersey
symstar -cg StarGrp buildcg -site London
```

The `setup` command and the `buildcg` command ignore BCV devices that you may have added to the composite group at the workload site. If remote BCVs are used to protect data during resynchronization of the synchronous and asynchronous target sites, you must manually add the BCVs to the synchronous and asynchronous composite groups.

Step 6: Adding BCV Devices to the SRDF/Star Configuration

Although optional, BCVs are strongly recommended at the synchronous and asynchronous target sites (i.e., NewJersey and London in Figure 4-2) because the act of starting a resynchronization activity between these sites temporarily compromises the consistency of the R2 data until resynchronization is fully completed. The BCVs retain a consistent restartable image of the data volumes during periods of resynchronization.

To add BCV devices to the SRDF/Star configuration, do the following:

1. Add BCVs at the remote target sites by associating the BCVs with the composite group:

   ```
   symbcv -cg StarGrp -sid 11 associateall dev -range 182:19A -rdf -rdfg 22
   symbcv -cg StarGrp -sid 11 associateall dev -range 3B6:3C9 -rdf -rdfg 23
   ```

   Note: Include the RDF group number of the local R1 source devices.

2. Synchronize the remote BCV pairs by issuing the following commands to copy data from the R2 devices on the remote Symmetrix arrays to the BCV devices there. The `-rdf` option signifies that the targets are the remote BCVs. The names `NewJersey` and `London` are those that were previously set for RDF groups 22 and 23, respectively. The `-star` option is required for any TimeFinder operations that affect BCV devices in an SRDF/Star composite group. To synchronize the remote BCV pairs, enter:

   ```
   symmir -cg StarGrp establish -star -full -rdf -rdfg name:NewJersey
   symmir -cg StarGrp establish -star -full -rdf -rdfg name:London
   ```

Note: BCVs can be added to a composite group either before or after performing the setup operation. The setup operation does not save BCV information for the composite group, so any BCV s that were associated are excluded from the internal definitions file which is copied to the remote hosts.

SRDF/Star Control Operations

Controlling SRDF/Star involves tasks such as bringing up the SRDF/Star sites for normal operation, isolating one of more sites for testing or other purposes, or switching the workload to one of the remote sites after primary site failure. You perform these and other SRDF/Star operations using the `symstar` commands that are listed in Table 4-1 and described in the following sections.

Table 4-1 symstar Control Operations

Control Operation Task	**`symstar`** Action Argument	Results	Workload (W) or Target (T) Task
Cleanup after a disaster (workload site) failure	`cleanup`	Cleans up internal meta information and Symmetrix cache at the remote site after a failure at the workload site.	T
Begin SRDF synchronization	`connect`	Starts the SRDF data flow in Adaptive Copy Disk mode.	W
Disable for full SRDF/Star protection	`disable`	Disables SRDF/Star consistency protection across the three sites.	W
Suspend SRDF synchronization	`disconnect`	Suspends the SRDF data flow and transitions the path to Adaptive Copy Disk mode.	W
Enable for full SRDF/Star protection	`enable`	Enables complete SRDF/Star consistency protection across the three sites.	W
Halt the SRDF/Star system	`halt`	Used to prepare the system for a planned switch of the workload to a target site. This action write-disables the R1 devices, drains all invalid tracks and MSC cycles so that SiteA=SiteB=SiteC, suspends RDF links, disables all consistency protection, and sets Adaptive Copy Disk mode.	W
Isolate a target site from the SRDF/Star configuration	`isolate`	Isolates one target site from the SRDF/Star configuration and makes its R2 devices read/write enabled to their hosts.	W
Enable SRDF consistency protection for a target site	`protect`	Synchronizes devices between the workload and target sites and enables SRDF/Star consistency protection to the specified target site.	W
Display SRDF/Star status	`query`	Displays the status of a given SRDF/Star site configuration.	W/T
Reset after a transient failure	`reset`	Cleans up internal meta information and Symmetrix cache at the remote site after transient fault (e.g. loss of connectivity to the synchronous or asynchronous target site).	W
Display the SRDF/Star internal definition	`show`	Displays the contents of the internal definition for a given SRDF/Star site configuration.	W/T
Switch worload operations to a target site	`switch`	Transitions workload operations to a target site after a workload site failure or as part of a planned event.	T
Disable SRDF consistency protection for a target site	`unprotect`	Disables SRDF/star consistency protection to the specified target site.	W

Note: To perform a `symstar` command, SRDF/Star needs to be in an allowable state. Otherwise, a message is returned, stating that SRDF/Star is not in a state that permits the particular operation that you are attempting to perform. Refer to Appendix B for a list of the allowable states for each SRDF/Star control operation. The System State can be displayed with the `symstar query` command.

Note: The examples provided in this section use the name `StarGrp` for the composite group and the names `NewYork`, `NewJersey`, and `London` as the names for the workload site, the synchronous target site, and the asynchronous target site, respectively (refer to Figure 4-2 on page 4-6).

Bringing Up the SRDF/Star Sites for Normal Operation

Once you have completed all of the setup tasks described in the previous section, *Setting up SRDF/Star*, you can bring up your SRDF/Star site configuration using a control host at the workload site. However, you first need to use the `symstar query` command to determine if the setup action left the target sites in a *Connected* or *Disconnected* state. For example:

```
symstar -cg StarGrp query -detail
```

For an example of the output returned with this command, refer to *Using the symstar show and query Commands* on page 4-13.

If the system state is *Connected*, the devices are already read/write (RW) on the RDF link, in which case you can omit the `symstar connect` commands from the command sequence below.

If the system state is *Disconnected*, the following command sequence brings up SRDF/Star by first bringing up site NewJersey and then site London. (To bring up site London first and then NewJersey, reverse the order in which you execute the `symstar protect` commands.

```
symstar -cg StarGrp connect -site NewJersey
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site NewJersey
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

Where:

The `connect` action sets the mode to Adaptive Copy Disk and brings the devices to RW on the RDF links, but does not wait for synchronization.

The `protect` action transitions to the correct SRDF mode (sync or async), enables RDF consistency protection, waits for synchronization, and sets the Star mode indicators.

The `enable` action provides complete SRDF/Star protection. It creates and initializes the SDDF resources, sets the Star mode indicators on recovery groups, and enables SRDF/Star so that it waits for R2-recoverable Star protection across SRDF/S and SRDF/A before producing a Star Protected state.

Using the symstar show and query Commands

You can use the `symstar show` command to display the contents of the internal SRDF/Star definition file that was created when the `symstar setup` command was executed. To display the internal definition file for a composite group called `StarGrp`, enter:

`symstar -cg StarGrp show`

```
CG name                                : StarGrp

Synchronous Target Site Information   :
  {
  --------------------------------------
  Source (R1) View      Target (R2) View
  ----------------  --------------------
               RD                RD
  Symmetrix     F   Symmetrix     F  Rcvy
  ID            G   ID            G  RDFG
  ------------ --   ------------ --   ----
  000190300150 37   000190300180 37    38
  000190300150 20   000190300180 20    36
  --------------------------------------
  }

Asynchronous Target Site Information :
  {
  --------------------------------------
  Source (R1) View      Target (R2) View
  ----------------  --------------------
               RD                RD
  Symmetrix     F   Symmetrix     F  Rcvy
  ID            G   ID            G  RDFG
  ------------ --   ------------ --   ----
  000190300150 38   000190300152 38    37
  000190300150 36   000190300152 36    10
  --------------------------------------
  }

Option file settings                   :
  {
  WorkloadSite                         : NewYork
  SyncTargetSite                       : NewJersey
  AsyncTargetSite                      : London
  Adaptive_Copy_Tracks                 : 30000
  Action_Timeout                       : 1800
  }
```

Note: To have the display include all of the devices with SRDF/Star, include the `-detail` option.

You can use the `symstar query` command to display the local and remote Symmetrix array information and the status of SRDF pairs in the composite group. To display the status of the SRDF/Star site configuration for a composite group called `StarGrp`, enter:

`symstar -cg StarGrp query`

```
Site Name                              : NewYork

Workload Site                          : NewYork
1st Target Site                        : NewJersey
2nd Target Site                        : London

Workload Data Image Consistent         : Yes
System State:
{
```

```
1st_Target_Site                                : Connected
2nd_Target_Site                                : Connected
}

Last Action Performed                          : Setup
Last Action Status                             : Successful
Last Action Timestamp                          : 02/03/2006_14:45:02

STAR Information:
{
STAR Consistency Capable                       : Yes
STAR Consistency Mode                          : NONE
Synchronous Target Site                        : NewJersey
Asynchronous Target Site                       : London
Differential Resync Available                  : N/A
R2 Recoverable                                 : N/A
  Asynchronous Target Site Data most Current   : N/A
}

1st Target Site Information:
  {
  Site Name                                    : NewJersey
  RDF Consistency Capability                   : SYNC
  RDF Consistency Mode                         : NONE
  Site Data Image Consistent                   : No

         Workload Site                    Target Site
----------------------------- -- ----------------------------- - ------------
         ST                    LI          ST                   M
      RD  A                     N Rem   RD  A                   O
Symm   F  T   R1 Inv   R2 Inv  K Symm   F  T   R1 Inv   R2 Inv D RDF Pair
ID     G  E   Tracks   Tracks  S ID     G  E   Tracks   Tracks E STATE
----- -- -- -------- -------- -- ----- -- -- -------- -------- - ------------
00150 51 RW        0        0 RW 00180 51 WD        0        0 S Synchronized
00150 53 RW        0        0 RW 00180 53 WD        0        0 S Synchronized
Totals:  -- -------- -------- --          -- -------- -------- - ------------
         RW        0        0 RW          WD        0        0 S Synchronized
}

2nd Target Site Information:
  {
  Site Name : London
  RDF Consistency Capability : MSC
  RDF Consistency Mode : NONE
  Site Data Image Consistent : No

         Workload Site                    Target Site
----------------------------- -- ----------------------------- - ------------
         ST                    LI          ST                   M
      RD  A                     N Rem   RD  A                   O
Symm   F  T   R1 Inv   R2 Inv  K Symm   F  T   R1 Inv   R2 Inv D RDF Pair
ID     G  E   Tracks   Tracks  S ID     G  E   Tracks   Tracks E STATE
----- -- -- -------- -------- -- ----- -- -- -------- -------- - ------------
00150 52 RW        0        0 RW 00152 52 WD        0        0 S Synchronized
00150 54 RW        0        0 RW 00152 54 WD        0        0 S Synchronized
Totals:  -- -------- -------- --          -- -------- -------- - ------------
         RW        0        0 RW          WD        0        0 S Synchronized
  }
Legend:
 Modes:
    Mode of Operation: A=Async, C=Adaptive Copy, S=Sync, O=Other, M=Mixed
```

Note: Using the `-detail` option with the `query` command will include extended information in the output.

Isolating SRDF/Star Sites

There may be occasions when you want to isolate one of the SRDF/Star sites, perhaps for testing purposes, and then join the isolated site again with the SRDF/Star configuration. The `symstar isolate` command allows you to temporarily isolate one or all SRDF/Star sites.

The `symstar isolate` command has the following requirements:

- SRDF/Star protection must be disabled.
- The site to be isolated must be in the *Protected* state.
- If there are BCVs at the target site that are paired with the SRDF/Star R2 devices, the BCV pairs should be split prior to executing the command.

Isolating a Protected Target Site

If SRDF/Star is running normally and in the *STAR Protected* state, the `symstar disable` command disables STAR but leaves both target sites in the *Protected* state, from which you can isolate either site. For example:

```
symstar -cg StarGrp disable
symstar -cg StarGrp isolate -site NewJersey
```

This action isolates site NewJersey by splitting its SRDF pairs and making the R2 devices read/write enabled to the NewJersey host.

Isolating a Disconnected Target Site

If the site you want to isolate is in the *Disconnected* state, you must first get it to the *Protected* state with the `connect` and `protect` commands. For example:

```
symstar -cg StarGrp connect -site NewJersey
symstar -cg StarGrp protect -site NewJersey
symstar -cg StarGrp isolate -site NewJersey
```

Rejoining an Isolated Site

After performing testing or other tasks in NewJersey that require the isolation, you can rejoin the NewJersey site with the SRDF/Star configuration and enable SRDF/Star protection again. To do this, you must first transition NewJersey from the *Isolated* state to the *Disconnected* state. Then proceed to connect and protect. For example:

```
symstar -cg StarGrp disconnect -site NewJersey
symstar -cg StarGrp connect -site NewJersey
symstar -cg StarGrp protect -site NewJersey
symstar -cg StarGrp enable
```

In rejoining an isolated site to the SRDF/Star configuration, any updates made to NewJersey’s R2 devices while *isolated* are discarded. That is, the data on the R1 devices overwrites the data on the R2 devices.

After rejoining the NewJersey site, you need to re-establish any NewJersey BCV pairs that are part of the `StarGrp` composite group.

Responding to Transient Remote Faults

An SRDF/Star fault caused by network or remote storage controller faults is considered a transient fault because it does not disrupt the production workload site. Only the transfer of data across the link is affected. Transient faults during normal SRDF/Star operation require a recovery action.[1] For example, a network fault has temporarily interrupted communication on the SRDF/A link (Figure 4-3). When the physical cause of the transient fault is corrected, you can execute a series of commands to clean up internal meta data and Symmetrix cache at the asynchronous target site and return the site to SRDF/Star protection.

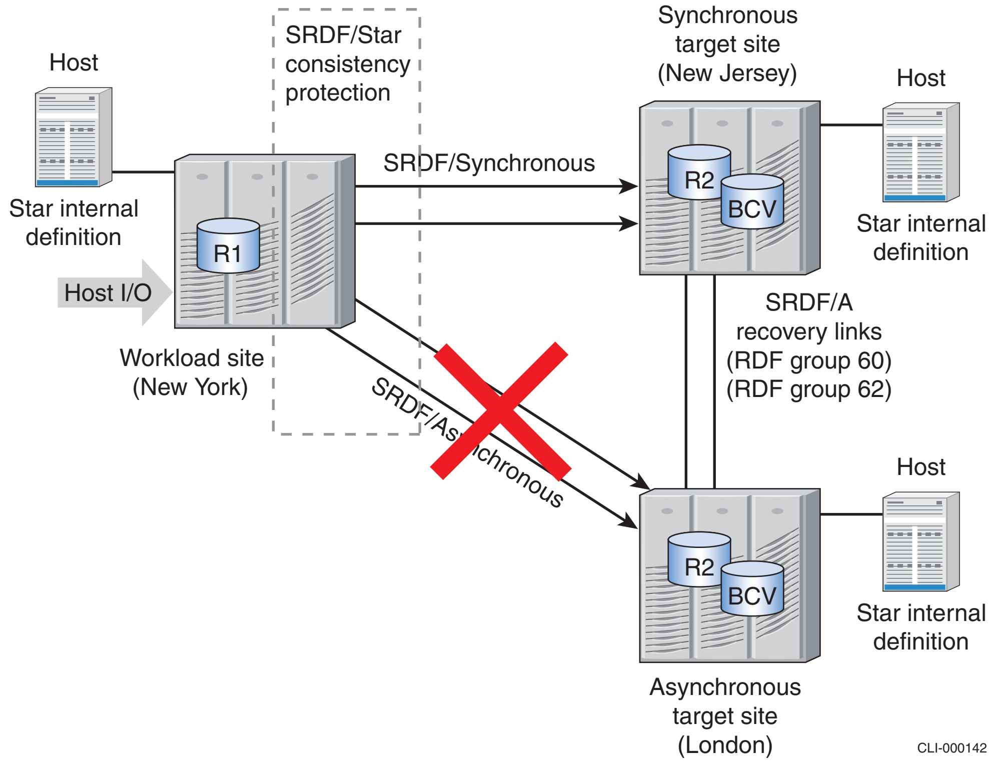

Figure 4-3 Transient Failure Recovery

The following procedure assumes that the transient fault affected the SRDF/A link between the NewYork site and the London site, the NewJersey state would still be *Protected*, while the London state would be *PathFail*.

1. Clean up any internal meta data or Symmetrix cache remaining at the London site after the transient fault occurred by entering:

   `symstar -cg StarGrp reset -site London`

1. The recovery action described in this section only applies if the transient fault occurs while the system is in the *Protected* or *Star_Protected* states. If a transient fault occurs on a link that is in the *Connected* state, the link is disconnected. Restarting synchronization again from a *Disconnected* state (after correcting the cause of the failure) would require only the `connect` action. A transient fault that occurs when the link is in a *Disconnected* state is not displayed.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the remote site, you should perform the appropriate TimeFinder action after executing the `reset` command.

   For example, to split off a consistent restartable image of the data volumes prior to the resynchronization process, enter:

   ```
   symmir -cg StarGrp split -star -rdf -rdfg name:London
   ```

   Note: Splitting the remote BCVs after a transient fault maintains a consistent image of the data at the remote site until that time when it is safe to re-establish the BCVs with the R2 devices. The act of starting the resynchronization activity will temporarily compromise the consistency of the R2 data until the resynchronization is fully completed. The split BCVs retain a consistent restartable image of the data volumes during periods of SRDF/Star resynchronization.

3. Perform the necessary composite actions required to return the London site to the SRDF/Star configuration by entering:

   ```
   symstar -cg StarGrp connect -site London
   symstar -cg StarGrp protect -site London
   symstar -cg StarGrp enable
   ```

4. If any London BCV pairs are part of the `StarGrp` composite group, re-establish them by entering:

   ```
   symmir -cg StarGrp establish -star -rdf -rdfg name:London
   ```

Responding to Disaster Faults

An SRDF/Star fault caused by the loss of the workload site is considered a disaster because it disrupts the production workload. Switching procedures allow to switch the workload to either of the remaining remote sites and resume data replication between those two sites.

For example, a disaster fault has caused the loss of the production workload site (Figure 4-4). You can execute a series of commands from a remote control host to clean up internal meta data and Symmetrix cache at the asynchronous target site, switch to one of the remote sites, and perform the resynchronization actions necessary to establish the new remote workload site.

If the workload site is lost, you can switch the production workload to either the synchronous or asynchronous target site. If the loss of the workload site was caused by a rolling disaster, the data at the synchronous target site can be ahead of the data at asynchronous site, or vice versa (as indicated by the `symstar query` command). You can specify which site data to keep in the `symstar switch` command.

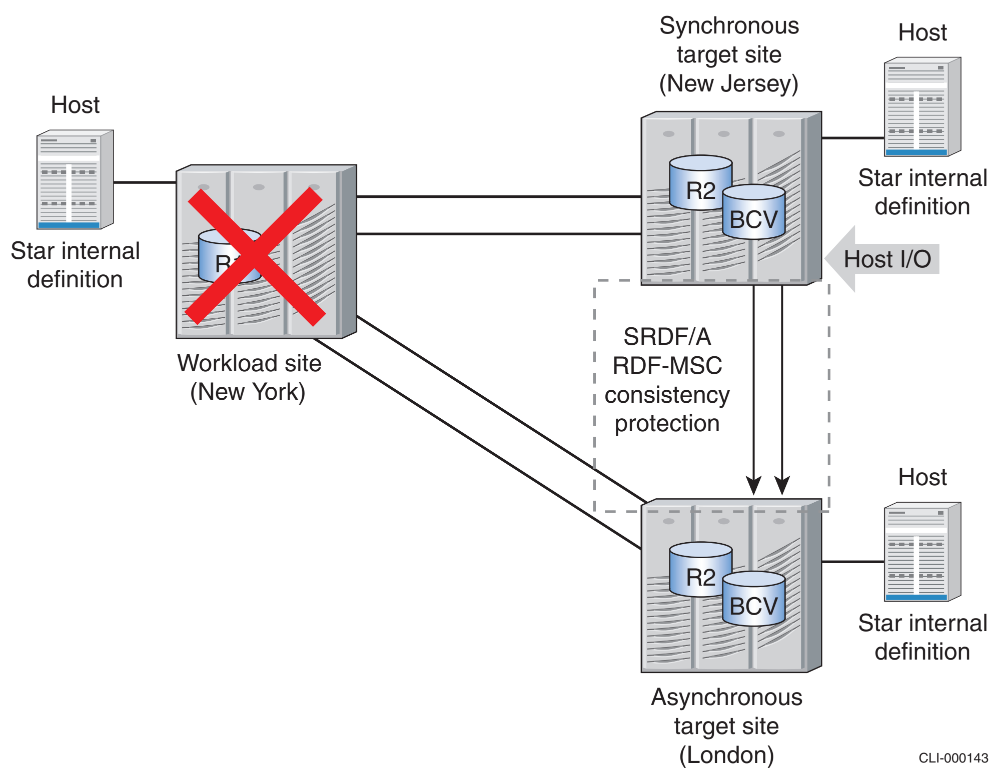

Figure 4-4 Loss of Workload Site and Recovery

Switching Workload to the Synchronous Target Site

The following procedure explains how to perform an unplanned switch operation, which means that the system state is 1st_target_site:*Pathfail*, 2nd_target_site:*Pathfail*, and STAR:*Tripped*. This procedure uses `symstar` commands from the NewJersey control host to bring up the synchronous site NewJersey as the new workload site. The NewJersey data will then be replicated asynchronously to the asynchronous target site (London).

Note: When switching the workload to the synchronous target site but choosing to keep the data from the asynchronous target site, there will be a wait for all the RDF data to synchronize before the application workload can be started at the synchronous site. The `symstar switch` command does not return control until the data is synchronized.

---

1. Clean up any internal meta data or Symmetrix cache remaining at the London site after the loss of the workload site occurred by entering:

   ```
   symstar -cg StarGrp cleanup -site London
   ```

   The `symstar cleanup` command cleans up internal meta data and Symmetrix cache after a failure. The cleanup action always applies to the asynchronous site. Since the target site in this case is London, this action performs the cleanup tasks there.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the London site, you should perform the appropriate TimeFinder action after executing the `cleanup` command. However, prior to performing a switch and resynchronization operation between NewJersey and London, there is no existing RDF relationship between the synchronous and asynchronous target sites, therefore the BCV control operation must be performed via a separate device file instead of the composite group. In this case, the device file (`StarFileLondon`) defines the BCV pairs on Symmetrix 13 in London.

   For example, to split off a consistent restartable image of the data volumes during the resynchronization process using the device file, enter:

   ```
   symmir -f StarFileLondon split -star -sid 13
   ```

   Note: Splitting the remote BCVs after a workload site failure maintains a consistent image of the data at the remote site until that time when it is safe to re-establish the BCVs with the R2 devices.

   ---

3. Use the `symstar switch` command to perform the tasks necessary to allow the workload to be started at the specified site (for example, specifying NewJersey as the new workload site and changing the R2 devices there into R1 devices). The following specifies to keep the NewJersey data instead of the London data:

   ```
   symstar -cg StarGrp switch -site NewJersey -keep_data NewJersey
   symstar -cg StarGrp connect -site London
   symstar -cg StarGrp protect -site London
   ```

   The `connect` and `protect` actions perform the tasks necessary to reconfigure the RDF devices between NewJersey and London into SRDF pairs (R1 devices at site NewJersey paired with the R2 devices at site London) and perform the differential resynchronization of the data between these sites. Once the recovery tasks are complete, the NewJersey production workload is remotely protected through a asynchronous link to London. You can begin the production workload at NewJersey any time after the `switch` action completes. However, doing it before the `connect` and `protect` actions complete means you have no remote protection for a period of time.

4. After recovering the London site, you need to re-establish any BCV pairs at the London site. You can either use the device file syntax (`-f StarFileLondon`) or, if you have associated the London BCV pairs with the StarGrp composite group on the control host, you can use the `-cg` syntax. To re-establish any London BCV pairs that are part of the StarGrp composite group, enter:

```
symmir -cg StarGrp establish -star -rdf -rdfg name:London
```

5. Once the NewYork site is repaired, you may want to bring NewYork back into the SRDF/Star system while keeping the production workload at site NewJersey. For example, to recover and enable the NewYork site, enter the following commands from the NewJersey control host:

```
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp enable
```

Switching Workload to Asynchronous Target Site

The following procedure explains how to perform an unplanned switch operation, which means that the system state is 1st_target_site:*Pathfail*, 2nd_target_site:*Pathfail*, and STAR:*Tripped*. This procedure uses `symstar` commands from the London control host to bring up the asynchronous site London as the new workload site. The London data will then be replicated asynchronously to the synchronous target site (NewJersey).

Note: When switching the workload to the asynchronous target site but choosing to keep the data from the synchronous target site, there will be a wait for all the RDF data to synchronize before the application workload can be started at the asynchronous site. The `symstar switch` command does not return control until the data is synchronized.

1. Clean up any internal meta data or Symmetrix cache remaining at the London site after the loss of the workload site occurred by entering:

```
symstar -cg StarGrp cleanup -site London
```

   The `symstar cleanup` command cleans up internal meta data and Symmetrix cache after a failure. The cleanup action always applies to the asynchronous site. Since the target site in this case is London, this action performs the cleanup tasks there.

2. If you are protecting SRDF/Star data with TimeFinder BCVs at the NewJersey site, you should perform the appropriate TimeFinder action after executing the `cleanup` command. However, prior to performing a switch and resynchronization operation between London and NewJersey, there is no existing RDF relationship between the synchronous and asynchronous target sites, so the BCV control operation must be performed via a separate device file instead of the composite group. In this case, the device file (`StarFileLondon`) defines the BCV pairs on Symmetrix 13 in London.

   For example, to split off a consistent restartable image of the data volumes during the resynchronization process using the device file, enter:

```
symmir -f StarFileLondon split -star -sid 13
```

   Note: Splitting the remote BCVs after a workload site failure maintains a consistent image of the data at the remote site until that time when it is safe to re-establish the BCVs with the R2 devices.

3. Use the `symstar switch` command to perform the tasks necessary to allow the workload to be started at the specified site (for example, specifying London as the new workload site and changing the R2 devices there into R1 devices). The following specifies to keep the NewJersey data instead of the London data:

   ```
   symstar -cg StarGrp switch -site London -keep_data NewJersey
   symstar -cg StarGrp protect -site NewJersey
   ```

   The `connect` action is not required here because the `switch` action must perform that function to get the remote data from the NewJersey site.

   Because London is using the NewJersey data, you cannot start the application workload in London until the `switch` action completes (the `symstar switch` command blocks further action until it completes). This ensures that all of the SRDF pairs are synchronized prior to starting the workload.

4. After recovering the NewJersey site, you need to re-establish any BCV pairs at the London site. You can either use the device file syntax (`-f StarFileNewJersey`) or, if you have associated the NewJersey BCV pairs with the `StarGrp` composite group on the control host, you can use the `-cg` syntax. To re-establish any NewJersey BCV pairs that are part of the `StarGrp` composite group, enter:

   ```
   symmir -cg StarGrp establish -star -rdf -rdfg name:NewJersey
   ```

5. Once the NewYork site is repaired, you cannot bring it back under SRDF/Star protection without switching the workload back to NewYork or NewJersey. If the workload remains at London, you can connect to NewYork by executing a `connect` action from the London control host. The connect action sets the mode to Adaptive Copy Disk and brings the devices to RW on the RDF links.

   ```
   symstar -cg StarGrp connect -site NewYork
   ```

6. With the workload at asynchronous site London, you can perform a `protect` action on NewYork only if you first unprotect NewJersey. From the distant site, only one link at a time can operate in Asynchronous mode. The protect action transitions the link from Adaptive Copy mode to Asynchronous mode and enable RDF consistency protection.

   The `symstar enable` action is blocked.

Note: To see expanded operational examples for SRDF/Star, you can refer to the following Technical Note, *Using SYMCLI to Implement SRDF/Star.*

Conducting Planned Switching Operations

A planned switch operation switches the workload function to one of the remote target sites, even though the original workload site is operating normally.

The system state is usually *Star Protected* (or the target sites are at least *Connected*) prior to starting a switch to one of the remote target sites. Use the `symstar query` command to confirm the system state.

Regardless of which remote site you are switching to, you must first stop the application workload at the current workload site, unmount the file systems, export volume groups, and so forth. Then execute the SRDF/Star `halt` action from the control host. For example:

```
symstar -cg StarGrp halt
```

Note: If you change your mind after halting the SRDF/Star system, issue the `halt -reset` command so that the workload can be restarted on the same host.

The `halt` action disables the R1 devices, waits for all invalid tracks and cycles to drain, suspends the RDF links, disables Consistency protection, and sets the Star mode indicators. This results in the target sites transitioning to the *Halted* state, and the data on all three sites being the same.

For example. when executed from the NewJersey control host, the following command sequence switches the workload to the synchronous target site and connects NewJersey to NewYork (synchronously), and NewJersey to London (asynchronously):

```
symstar -cg StarGrp switch -site NewJersey
symstar -cg StarGrp connect -site NewYork
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site NewYork
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

Switching the Workload from a Target Site Back to the Original Workload Site

After a planned or unplanned switch of the workload to the synchronous or asynchronous target site, you can (if the original workload site is operating normally) with production operations back to the original workload site to re-establish the original SRDF/Star configuration.

To switch back to the original workload site, you must be able to completely synchronize the data at all three sites. Before initiating a switch back to the original workload site, the current workload site’s RDF links must be connected to the other two sites. The following states allow you to switch from the synchronous target site to the original workload site:

- *Star Protected*
- Both target sites are *Protected*
- One target site is *Protected* and the other is *Connected*
- Both target sites are *Connected*

The following states allow you to switch from the asynchronous target site to the original workload site:

- One target site is Protected and the other is Connected
- Both target sites are Connected

Regardless of which remote site you are working from, you must first stop the workload at that site and execute the halt action from the control host. Assuming the workload is currently running at NewJersey, issue the following command from the NewJersey control host:

```
symstar -cg StarGrp halt
```

The halt action disables the R1 devices, waits for all invalid tracks and cycles to drain, suspends the RDF links, disables RDF Consistency protection, and sets the Star mode indicators. This results in the target sites transitioning to the *Halted* state, and all the data on all three sites being the same.

This example assumes that the workload was running at the synchronous site NewJersey before stopping it and performing the `halt` action. You must execute the following command sequence from the NewYork control host. The resulting actions switch the workload to NewYork and reconnect NewYork to NewJersey (synchronously), and NewYork to London (asynchronously).

```
symstar -cg StarGrp switch -site NewYork
symstar -cg StarGrp connect -site NewJersey
symstar -cg StarGrp connect -site London
symstar -cg StarGrp protect -site NewJersey
symstar -cg StarGrp protect -site London
symstar -cg StarGrp enable
```

The resulting state is *Star Protected*.

Disabling SRDF/Star for Device Reconfiguration

To reconfigure the devices in the SRDF/Star composite group (for example, different devices), you must return the SRDF/Star system to a pre-setup condition, from which you can then revise and rebuild the composite group and perform setup tasks as described earlier.

There are two ways to return the SRDF/Star system to a pre-setup condition:

- By unprotecting the synchronous and asynchronous target sites (the faster method)
- By disconnecting the synchronous and asynchronous target sites (to create consistent copies)

Unprotecting the Target Sites

To unprotect the target sites, you must first turn off SRDF/Star protection (assuming the system state is *Star Protected)*. The following command sequence must be executed from the workload site:

```
symstar -cg StarGrp disable
symstar -cg StarGrp unprotect -site NewJersey
symstar -cg StarGrp unprotect -site London
symstar -cg StarGrp setup -options options.StarGrp -clear
```

Where:

The `disable` action disables SRDF/Star protection and terminates the Star SDDF sessions.

The `unprotect` action disables RDF Consistency protection and sets the Star mode indicators.

The `setup -clear` action cleans up metadata.

Disconnecting the Target Sites

The alternative method of disconnecting the target sites instead of unprotecting them allows you to achieve the same results while, at the same time, creating a consistent copy of the data at each site:

```
symstar -cg StarGrp disable
symstar -cg StarGrp disconnect -site NewJersey
symstar -cg StarGrp disconnect -site London
symstar -cg StarGrp setup -options options.StarGrp -clear
```

PART II

Operational Examples

The Operational Examples part of this product guide identifies and focuses on some specific SRDF tasks that represent the most typical practices in the management of your Symmetrix storage environment. These practical examples illustrate various SDRF processes by showing the SYMCLI command sequences to accomplish these tasks. These specific management tasks are described in the subsequent chapters as follows:

Chapter 5, *Performing SRDF Control Operations*, provides examples of the SRDF control operations used to manage devices within various remote SRDF configurations.

Chapter 6, *Implementing Consistency Protection Using RDF-ECA and RDF-MSC*, provides examples for implementing consistency protection across one or more database management systems within an SRDF configuration using RDF Enginuity Consistency Assist (RDF-ECA) for synchronous mode and RDF Multi Session Consistency (RDF-MSC) for asynchronous mode.

Chapter 7, *Implementing Consistency Protection Using PowerPath*, provides examples for implementing consistency protection across one or more database management systems within an SRDF configuration using PowerPath.

Chapter 8, *Performing SRDF/Automated Replication Operations*, provides examples for replicating data in pre-defined cycles using the SRDF automated replication process.

Chapter 9, *Querying and Verifying with SRDF Commands*, provides examples on using the query and verify operations with SRDF family products.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

5

Performing SRDF Control Operations

This chapter provides examples of the Symmetrix command line interface (SYMCLI) actions and specific commands, which are used to manage devices within various SRDF configurations.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

Example 1: Basic SRDF Control Operations

The hardware setup for the following examples consists of two hosts, one connected to a local (source) Symmetrix and the other connected to a remote (target) Symmetrix. Display outputs may vary slightly according to the version of Solutions Enabler that you are using.

- The following `symrdf list` command from the local host displays information about local (R1) and remote (R2) SRDF devices. Entries in the RDF Typ:G column identify the device as either an R1 or R2 device and the RDF (RA) group number after the colon. The ellipsis (......) represents truncated output.

symrdf list

```
Symmetrix ID: 000000003264

                              Local Device View
-------------------------------------------------------------------------
                   STATUS    MODES                      RDF  S T A T E S
Sym         RDF    --------- -----  R1 Inv   R2 Inv ----------------------
Dev  RDev   Typ:G  SA RA LNK  MDA   Tracks   Tracks Dev RDev Pair
---- ----  ------ --------- ----- -------  ------- --- ---- -------------

0045 0045   R2:2  RW WD NR   S..        0    49500 WD  RW   Suspended
0046 0046   R2:2  ?? WD NR   S..        0    33000 WD  RW   Suspended
0047 0047   R2:2  ?? WD NR   S..        0        0 WD  RW   Suspended
009C 0054   R1:2  RW RW RW   S..        0        0 RW  NR   Synchronized
009D 0055   R1:2  RW RW RW   S..        0        0 RW  NR   Synchronized
009E 0056   R1:2  RW RW RW   S..        0        0 RW  NR   Synchronized
009F 0057   R1:2  RW RW RW   S..        0        0 RW  NR   Synchronized
00A0 0058   R1:2  RW RW RW   A.W        0        0 RW  NR   Synchronized
00A1 0059   R1:2  RW RW RW   A.W        0        0 RW  NR   Synchronized
00A2 005A   R1:2  RW RW RW   A.W        0        0 RW  NR   Synchronized
00A3 005B   R1:2  RW RW RW   A.W        0        0 RW  NR   Synchronized
```

- The following `symdev list` command from the local host with the `-r1` option displays all R1 devices. Those R1 devices that are not already part of a device group are displayed as "N/Grp'd," which means they are available to be added to a new RDF1 device group.

symdev list -r1

```
Symmetrix ID: 000000003264

        Device Name           Directors                 Device
---------------------------- ------------------------------------------
                                                                   Cap
Sym  Physical               SA :P DA :IT Config       Attribute    Sts   (MB)
---------------------------- ------------ ---------------------------------

009C /dev/rdsk/emcpower84c   16B:1 01A:C0 RDF1         N/Grp'd      RW     516
009D /dev/rdsk/emcpower85c   16B:1 02B:D3 RDF1         N/Grp'd      RW     516
009E /dev/rdsk/emcpower90c   16B:1 02A:C0 RDF1         N/Grp'd      RW     516
009F /dev/rdsk/emcpower91c   16B:1 01B:D3 RDF1         N/Grp'd      RW     516
00A0 /dev/rdsk/emcpower92c   16B:1 01B:C0 RDF1         N/Grp'd      RW     516
00A1 /dev/rdsk/emcpower93c   16B:1 02A:D3 RDF1         Grp'd        RW     516
00A2 /dev/rdsk/emcpower94c   16B:1 02B:C0 RDF1         N/Grp'd      RW     516
```

- Creating a device group and adding devices to it are prerequisites for performing SRDF operations.

  The following `symdg create` command from the local host creates a device group (Rdf1Grp). The `symld add` commands add standard devices to the group, using either a device's physical device (pd) name or, as shown below, its Symmetrix device (dev) name. In the `symdg show` display, "Device Group RDF Information" refers to information that is applicable to all RDF standard devices in the group.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp -sid 3264 add dev 9C
symld -g Rdf1Grp -sid 3264 add dev 9D
symdg show Rdf1Grp

Group Name:  Rdf1Grp

    Group Type                                    : RDF1
    Device Group in GNS                           : Yes
    Valid                                         : Yes
    Symmetrix ID                                  : 000000003264
    Group Creation Time                           : Tue Jan  6 12:08:17 2004
    Vendor ID                                     : EMC Corp
    Application ID                                : SYMCLI

    Number of STD Devices in Group                :    2
    Number of Associated GK's                     :    0
    Number of Locally-associated BCV's            :    0
    Number of Locally-associated VDEV's           :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (2):
        {
        --------------------------------------------------------------------
                                                       Sym               Cap
        LdevName              PdevName                 Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c2t6d3s2       009C      RW       516
        DEV002                /dev/rdsk/c2t6d4s2       009D      RW       516
        }

    Device Group RDF Information
        {
        RDF Type                                  : R1
        RDF (RA) Group Number                     : 2                (01)

        Remote Symmetrix ID                       : 000000003265

        R2 Device Is Larger Than The R1 Device : False

        RDF Mode                                  : Synchronous
        RDF Adaptive Copy                         : Disabled
        RDF Adaptive Copy Write Pending State    : N/A
        RDF Adaptive Copy Skew (Tracks)           : 65535

        RDF Device Domino                         : Disabled

        RDF Link Configuration                    : Fibre
        RDF Link Domino                           : Disabled
        Prevent Automatic RDF Link Recovery       : Disabled
        Prevent RAs Online Upon Power ON          : Enabled

        Device RDF Status                         : Ready            (RW)

        Device RA Status                          : Ready            (RW)
```

```
        Device Link Status                        : Ready           (RW)

        Device Suspend State                      : N/A
        Device Consistency State                  : Disabled
        RDF R2 Not Ready If Invalid               : Enabled

        Device RDF State                          : Ready           (RW)
        Remote Device RDF State                   : Not Ready       (NR)

        RDF Pair State (  R1 <===> R2 )           : Synchronized

        Number of R1 Invalid Tracks               : 0
        Number of R2 Invalid Tracks               : 0
        }
```

- When EMC installs an SRDF configuration, the installers usually establish static SRDF pairs at that time. The `symrdf query` command demonstrates the state of the SRDF devices and their RDF links. Under normal circumstances, the SRDF pair is synchronized (as shown below). The R1 devices are read-writeable and the RDF links are read-writeable. However, the R2 devices, which are acting as mirrors to the R1 devices, are write disabled (WD) and cannot be written to by the remote-site host at this time. The link is operating with Synchronous replication (indicated by an S in the M column).

symrdf -g Rdf1Grp query

```
Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
            ST                  LI      ST
Standard     A                   N       A
Logical      T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev  E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002  009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized

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

- While the identity of the remote SRDF devices of each pair is known (9C is paired with 54, and 9D with 55), the configuration of remote Symmetrix arrays connected to the local Symmetrix may not be known. You can usually determine the identity of these remote Symmetrix arrays using the `symcfg list` command shown below. However, to identify the remote Symmetrix that contains a specific R2 device, you need to issue a `symdev show` *`DeviceName`* command from the local host on its paired R1.

```
symcfg list
                                S Y M M E T R I X

                                        Mcode    Cache      Num Phys  Num Symm
    SymmID       Attachment  Model     Version  Size (MB)  Devices   Devices

    000000003264 Local       DMX2000P  5669       20480      100        396
    000000003263 Remote      DMX2000P  5669       20480        0        534
    000000003265 Remote      8230      5568       16384        0        504
```

- When two or more remote Symmetrix arrays are present, `symdev show` for a specific R1 device (9C) identifies its configured R2 device (54) and the remote Symmetrix on which it resides (000000003265).

```
symdev show 9C

Symmetrix ID: 000000003264

    Device Physical Name     : /dev/rdsk/c2t6d3s2

    Device Symmetrix Name    : 009C
    Device Serial ID         : 6409C321
    Symmetrix ID             : 000000003264

    Device Group Name        : Rdf1Grp
    Device Logical Name      : DEV001

    Attached BCV Device      : N/A

    Vendor ID                : EMC
    Product ID               : SYMMETRIX
    Product Revision         : 5669

    Device Emulation Type    : FBA
    Device Defined Label Type: N/A
    Device Defined Label     : N/A

    Device Block Size        : 512

    Device Capacity
        {
        Cylinders            :       1100
        Tracks               :      16500
        512-byte Blocks      :    1056000
        MegaBytes            :        516
        KiloBytes            :     528000
        }

    Device Configuration     : RDF1
```

```
    RDF Information
        {
        Device Symmetrix Name                  : 009C
        RDF Type                               : R1
        RDF (RA) Group Number                  : 2                 (01)

        Remote Device Symmetrix Name           : 0054
        Remote Symmetrix ID                    : 000000003265
```

- Another useful command to examine Symmetrix connections is `symcfg list -ra all`. From the local host, this command reaches all Symmetrix arrays (one or two hops away) accessible through RDF links and displays the Remote Link Director information. Information in the Remote Symm ID column below shows that both Symmetrix 3264 and 3263 are connected to 3265, but 3263 and 3264 are not connected to each other. Refer to *Example 4: Creating a Dynamic RDF Group* on page 5-40 for more information about this display.

`symcfg list -ra all`

```
Symmetrix ID: 000000003264 (Local)

                S Y M M E T R I X   R D F   D I R E C T O R S

                                                Remote        Local    Remote
  Ident   Symb   Num  Slot  Type        Attr  SymmID        RA Grp   RA Grp   Status

  RF-3A    03A     3     3  RDF-BI-DIR   -    000000003265  2 (01)   2 (B)   Online
  RF-3B    03B    19     3  RDF-BI-DIR   -    000000003265  2 (01)   2 (B)   Online

Symmetrix ID: 000000003263 (Remote)

                S Y M M E T R I X   R D F   D I R E C T O R S

                                                Remote        Local    Remote
  Ident   Symb   Num  Slot  Type        Attr  SymmID        RA Grp   RA Grp   Status

  RF-3A    03A     3     3  RDF-BI-DIR   -    000000003265  1 (00)   1 (A)   Online
  RF-3B    03B    19     3  RDF-BI-DIR   -    000000003265  1 (00)   1 (A)   Online

Symmetrix ID: 000000003265 (Remote)

                S Y M M E T R I X   R D F   D I R E C T O R S

                                                Remote        Local    Remote
  Ident   Symb   Num  Slot  Type        Attr  SymmID        RA Grp   RA Grp   Status

  RF-3A    03A     3     3  RDF-BI-DIR   -    000000003263  1 (A)    1 (00) Online
  RF-14A   14A    14    14  RDF-BI-DIR   -    000000003264  2 (B)    2 (01) Online
  RF-3B    03B    19     3  RDF-BI-DIR   -    000000003263  1 (A)    1 (00) Online
  RF-3B    03B    30    14  RDF-BI-DIR   -    000000003264  2 (B)    2 (01) Online
```

- The view from the remote host and target Symmetrix 3265 reflects the view from the source. Symmetrix 3265 shows up as local, whereas 3264 shows up as remote.

`symcfg list`

```
                                S Y M M E T R I X

                                          Mcode    Cache      Num Phys  Num Symm
    SymmID       Attachment  Model        Version  Size (MB)  Devices   Devices

    000000003265 Local       8230         5568       16384        58       504
    000000003263 Remote      DMX2000P     5669       20480         0       534
    000000003264 Remote      DMX2000P     5669       20480         0       396
```

- The `symrdf list` command issued from the remote host shows 54 and 55 as local, and 9C and 9D as remote. Note the RDF Typ:G column for SRDF device 47. The B- indicates an RDF BCV device, and R1 indicates an RDF1 type device. The G column value indicates that the device belongs to RDF group number 2.

`symrdf list`

```
Symmetrix ID: 000000003265

                              Local Device View
---------------------------------------------------------------------------
                   STATUS    MODES                         RDF  S T A T E S
Sym        RDF     --------- -----  R1 Inv   R2 Inv ----------------------
Dev  RDev  Typ:G   SA RA LNK  MDA   Tracks   Tracks Dev RDev Pair
---- ----  ------ --------- ----- -------  ------- --- ---- -------------

0047 0047 B-R1:2   ?? RW NR   S..         0        0 RW  WD   Suspended
0054 009C   R2:2   RW NR RW   S..         0        0 NR  RW   Synchronized
0055 009D   R2:2   RW NR RW   S..         0        0 NR  RW   Synchronized
0056 009E   R2:2   WD NR RW   S..         0        0 NR  RW   Synchronized
0057 009F   R2:2   WD NR RW   S..         0        0 NR  RW   Synchronized
```

- To issue the same SRDF commands from the remote-site host as from the local-site host, it is necessary to build an RDF2 remote-site device group that has the same definitions as the RDF1 local-site device group. The `symdg export` command creates a text file (Rdf1Grp.txt) that contains the RDF1 group definitions. You then use `rcp` (or `ftp`) to transfer that file to the remote host.

```
symdg export Rdf1Grp -f Rdf1Grp.txt -rdf
rcp Rdf1Grp.txt api28:/.
```

- On the remote host, the `symdg import` command builds the RDF2 device group using the definitions from the text file.

```
symdg import Rdf2Grp -f Rdf1Grp.txt

Adding standard device 0054 as DEV001...
Adding standard device 0055 as DEV002...
```

- The following `symld list` command from the remote host displays the new RDF2 device group.

`symld -g Rdf2Grp list`

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Standard Device Name          Directors            Device
-----------------------------------  ------------ -------------------------
                                                                       Cap
Logical      Physical           Sym   SA :P DA :IT Config       Att Sts  (MB)
-----------------------------------  ------------ -------------------------

DEV001       /dev/rdsk/c1t3d0s2 0054  15A:0 01A:C0 RDF2             NR     516
DEV002       /dev/rdsk/c1t3d1s2 0055  15A:0 02B:D3 RDF2             NR     516
```

- The following query from the remote host displays the status of device group Rdf2Grp, and this information is the same as the previous query from the local host. The link is operating with Synchronous replication, and the state of the R2 devices is Write Disabled (WD).

```
symrdf -g Rdf2Grp query

Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
            ST                  LI      ST
Standard     A                   N       A
Logical      T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 WD       0        0 RW 009C RW       0        0 S..   Synchronized
DEV002  0055 WD       0        0 RW 009D RW       0        0 S..   Synchronized

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

- The following `newfs` commands from the local host prepare the R1 devices for writing by creating a new file system on each. The physical device names for the R1 devices are c2t6d3s2 and c2t6d4s2 (refer back to the section where Rdf1Grp was created).

```
newfs /dev/rdsk/c2t6d3s2

newfs: construct a new file system /dev/rdsk/c2t6d3s2: (y/n)? y
/dev/rdsk/c2t6d3s2:     1054080 sectors in 1098 cylinders of 15 tracks, 64
   sectors
        514.7MB in 69 cyl groups (16 c/g, 7.50MB/g, 3584 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 15456, 30880, 46304, 61728, 77152, 92576, 108000, 123424, 138848,
   154272,
 169696, 185120, 200544, 215968, 231392, 245792, 261216, 276640, 292064,
 307488, 322912, 338336, 353760, 369184, 384608, 400032, 415456, 430880,
 446304, 461728, 477152, 491552, 506976, 522400, 537824, 553248, 568672,
 584096, 599520, 614944, 630368, 645792, 661216, 676640, 692064, 707488,
 722912, 737312, 752736, 768160, 783584, 799008, 814432, 829856, 845280,
 860704, 876128, 891552, 906976, 922400, 937824, 953248, 968672, 983072,
 998496, 1013920, 1029344, 1044768,

newfs /dev/rdsk/c2t6d4s2

newfs: construct a new file system /dev/rdsk/c2t6d4s2: (y/n)? y
/dev/rdsk/c2t6d4s2:     1054080 sectors in 1098 cylinders of 15 tracks, 64
   sectors
        514.7MB in 69 cyl groups (16 c/g, 7.50MB/g, 3584 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 15456, 30880, 46304, 61728, 77152, 92576, 108000, 123424, 138848,
   154272,
 169696, 185120, 200544, 215968, 231392, 245792, 261216, 276640, 292064,
 307488, 322912, 338336, 353760, 369184, 384608, 400032, 415456, 430880,
 446304, 461728, 477152, 491552, 506976, 522400, 537824, 553248, 568672,
 584096, 599520, 614944, 630368, 645792, 661216, 676640, 692064, 707488,
```

```
722912, 737312, 752736, 768160, 783584, 799008, 814432, 829856, 845280,
860704, 876128, 891552, 906976, 922400, 937824, 953248, 968672, 983072,
998496, 1013920, 1029344, 1044768,
```

- The following commands from the local host create two mount points for the two volumes, mount the first one, and create a file on it called firstfile.

```
mkdir /R1-1 /R1-2
mount /dev/rdsk/c2t6d3s2 /R1-1
touch /R1-1/firstfile
ls -l /R1-1/firstfile

-rw-r--r--   1 root     other          0 Apr 16 13:18 /R1-1/firstfile

umount /R1-1
```

- The following command from the local host splits the SRDF pairs in the device group. As part of the `symrdf split` command, the individual operations suspend and rw_enable r2 are performed. When the split is complete, a query will reveal the altered state of the links and the R2 devices.

```
symrdf -g Rdf1Grp -noprompt split

An RDF 'Split' operation execution is in progress for device group 'Rdf1Grp'.
   Please wait...

    Suspend RDF link(s).......................................Done.
    Read/Write Enable device(s) on RA at target (R2)..........Done.

The RDF 'Split' operation successfully executed for device group 'Rdf1Grp'.
```

- A query from the local host reveals that the links have been logically set to NR (not ready) and the state of the R2 devices has been changed from WD to RW.

```
symrdf -g Rdf1Grp query

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K         T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev     E  Tracks   Tracks MDA    STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 NR 0054 RW       0        0 S..   Split
DEV002  009D RW       0        0 NR 0055 RW       0        0 S..   Split

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

- The following commands from the remote host create two mount points on the target and examine the contents of device 54 (c1t3d0s2) to confirm the existence of the file called firstfile.

```
mkdir /R2-1 /R2-2
mount /dev/rdsk/c1t3d0s2 /R2-1
ls -l /R2-1

total 16
-rw-r--r--    1 root     other           0 Apr 16 13:18 firstfile
drwx------    2 root     root         8192 Apr 16 13:13 lost+found
```

- While the local and remote site are split, both R1 and R2 devices are accessible for reads and writes. The following commands from the remote host change the contents of the R2 device by deleting firstfile on the target side and replacing it with a file called secondfile.

```
rm /R2-1/firstfile; touch /R2-1/secondfile
ls -l /R2-1

total 16
drwx------    2 root     root         8192 Apr 16 13:13 lost+found
-rw-r--r--    1 root     other           0 Apr 16 14:17 secondfile

umount /dev/rdsk/c1t3d0s2
```

- The `symrdf query` displays the results of changing the contents of the R2 device — that there are now local (R1) invalid tracks on the target (R2) side.

```
symrdf -g Rdf2Grp query

Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI       ST
Standard      A                    N        A
Logical       T  R1 Inv   R2 Inv  K         T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev     E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW       3         0 NR 009C RW       0         0 S..   Split
DEV002  0055 RW       0         0 NR 009D RW       0         0 S..   Split

Total            -------- --------            -------- --------
  Track(s)              3         0                   0         0
  MB(s)               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
```

- The following command from the remote site performs an incremental establish for the SRDF pairs in device group Rdf2Grp, copying to the R2 device any changes that have been made to the R1 device while the devices were split. Like all RDF control operations, you can initiate the establish action from either the local or remote site with the same results. The individual operations that combine to create an establish action are logged as they occur. For a more detailed report, you can examine the log file in `/var/symapi/log/symapi-yyyymmdd.log`.

```
symrdf -g Rdf2Grp -noprompt establish

An RDF 'Incremental Establish' operation execution is in progress for
   device group 'Rdf2Grp'. Please wait...

    Write Disable device(s) on RA at target (R2).............Done.
    Suspend RDF link(s)......................................Done.
    Mark target (R2) devices to refresh from source (R1)......Started.
    Device: 0054 ........................................... Marked.
    Mark target (R2) devices to refresh from source (R1)......Done.
    Suspend RDF link(s)......................................Done.
    Merge device track tables between source and target.......Started.
    Device: 009C ........................................... Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s).......................................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   device group 'Rdf2Grp'.
```

- The `symrdf verify` command from the remote host confirms that the SRDF pairs are completely synchronized.

```
symrdf -g Rdf2Grp verify

All devices in the RDF group 'Rdf2Grp' are in the 'Synchronized' state.
```

- A `symrdf split` command from the remote host followed by an examination of device `c1t3d0s2` confirms that the recently created secondfile on the R2 device has been removed and firstfile has been restored as a result of re-establishing the SRDF device pair.

```
symrdf -g Rdf2Grp -noprompt split

An RDF 'Split' operation execution is in progress for device group 'Rdf2Grp'.
Please wait...

    Suspend RDF link(s)......................................Done.
    Read/Write Enable device(s) on RA at target (R2).........Done.

The RDF 'Split' operation successfully executed for device group 'Rdf2Grp'.

mount /dev/rdsk/c1t3d0s2 /R2-1
ls -l  /R2-1

total 16
-rw-r--r--   1 root     other          0 Apr 16 13:18 firstfile
drwx------   2 root     root        8192 Apr 16 13:13 lost+found
```

- In preparation for demonstrating a restore operation, the following commands from the remote host replace firstfile on the R2 device with a file called thirdfile.

```
rm /R2-1/firstfile; touch /R2-1/thirdfile
ls -l /R2-1

total 16
drwx------   2 root     root        8192 Apr 16 13:13 lost+found
-rw-r--r--   1 root     other          0 Apr 16 14:56 thirdfile

umount /R2-1
```

- The `symrdf query` from the local host displays again the results of changing the contents of the R2 device — that there are now local (R1) invalid tracks on the target (R2) side.

```
symrdf -g Rdf1Grp query
Device Group (DG) Name: Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW        0        0 NR 0054 RW       3        0 S..   Split
DEV002  009D RW        0        0 NR 0055 RW       0        0 S..   Split

Total          -------- --------           -------- --------
  Track(s)            0        0                  3        0
  MB(s)             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
```

- The following `symrdf restore` command is issued from the local host on the source side. Because the `-full` option is omitted from the command line, the system performs an incremental restore, copying tracks that changed on the R2 device to the R1 device. In the process, any tracks on the R1 side that changed while the SRDF pairs were split are overwritten with data from corresponding tracks on the R2 side. When the restore is complete, the R1 device will contain the same data as the R2 device.

```
symrdf -g Rdf1Grp -noprompt restore

An RDF 'Incremental Restore' operation execution is in progress for device
   group 'Rdf1Grp'. Please wait...

    Write Disable device(s) on SA at source (R1)..............Done.
    Write Disable device(s) on RA at target (R2)..............Done.
    Suspend RDF link(s).......................................Done.
    Merge device track tables between source and target.......Started.
    Devices: 009C-009D ...................................... Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s)........................................Done.
    Read/Write Enable device(s) on SA at source (R1)..........Done.

The RDF 'Incremental Restore' operation successfully initiated for device
   group 'Rdf1Grp'.
```

- The following commands from the local host mount the c2t6d3s2 device (an R1 device) and examine its contents. The directory listing below confirms that the restore operation copied thirdfile from the R2 device to the R1 device.

```
mount /dev/rdsk/c2t6d3s2 /R1-1
ls -l /R1-1

total 16
drwx------   2 root     root        8192 Apr 16 13:13 lost+found
-rw-r--r--   1 root     other          0 Apr 16 14:56 thirdfile
```

◆ The following query from the local host illustrates that the SRDF pairs are now in the in the Synchronized state. (Note that the same restore operation with TimeFinder/Mirror software places the standard device in the Restored state. However, SRDF does not use the Restored state and places SRDF pairs in the Synchronized state after either an establish or restore operation.)

symrdf -g Rdf1Grp query

```
Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002  009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized

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

◆ A failover operation from the local host is similar to a split operation. However, because a failover is usually executed when a disaster on the source side necessitates moving data processing to the target side, a failover will write disable the R1 devices.

symrdf -g Rdf1Grp -noprompt failover

```
An RDF 'Failover' operation execution is in progress for device group
   'Rdf1Grp'.
   Please wait...

     Write Disable device(s) on SA at source (R1)..............Done.
     Suspend RDF link(s).......................................Done.
     Read/Write Enable device(s) on RA at target (R2)..........Done.

The RDF 'Failover' operation successfully executed for device group
   'Rdf1Grp'.
```

- The following query from the local host shows that the R1 devices are write disabled (WD), the RDF links have been suspended, and the R2 devices are read/write enabled (RW).

symrdf -g Rdf1Grp query

```
Device Group (DG) Name: Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K         T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------
DEV001  009C WD        0        0 NR 0054 RW        0        0 S..   Failed Over
DEV002  009D WD        0        0 NR 0055 RW        0        0 S..   Failed Over

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

- While data processing by the remote host continues on the target (R2) side, Symmetrix keeps a record of the tracks on the R2 side that have changed since the failover. The remote (R1) invalid tracks on the target (R2) side are those tracks that must be copied from the R2 device to the R1 device when the RDF links are re-established and a failback is performed. For the example, a C-Shell interactive script is run to continually rewrite the data on the R2 devices. The subsequent query from the remote host illustrates that there is a continuous accumulation of remote (R1) invalid tracks on the target (R2) side.

```
while (1)
   ? dd if=/dev/rdsk/c1t3d0s2 of=/dev/rdsk/c1t3d0s2 bs=1024k count=512
   ? dd if=/dev/rdsk/c1t3d1s2 of=/dev/rdsk/c1t3d1s2 bs=1024k count=512
   ? end
```

symrdf -g Rdf2Grp query

```
Device Group (DG) Name: Rdf2Grp
DG's Type              : RDF2
DG's Symmetrix ID      : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K         T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW   16385        0 NR 009C WD        0        0 S..   Failed Over
DEV002  0055 RW   16385        0 NR 009D WD        0        0 S..   Failed Over

Total               -------- --------           -------- --------
  Track(s)             32770        0                  0        0
  MB(s)               1024.0      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
```

- While the R2 side remains accessible for reads and writes, the `symrdf update` command from the local host takes a one-time snapshot of the remote (R1) invalid tracks on the target (R2) side for each device in the group (16385 in each case) and copies those tracks to the R1 side. The function of the update operation is to minimize downtime when issuing a failback command, which write disables the R2.

  `symrdf -g Rdf1Grp -noprompt update`

```
An RDF 'Update R1' operation execution is in progress for device group
   'Rdf1Grp'.
   Please wait...

     Suspend RDF link(s).......................................Done.
     Merge device track tables between source and target.......Started.
     Device: 009C ........................................... Merged.
     Device: 009D ........................................... Merged.
     Merge device track tables between source and target.......Done.
     Resume RDF link(s)........................................Done.

The RDF 'Update R1' operation successfully initiated for device group
   'Rdf1Grp'.
```

- The `symrdf query` command from the remote host shows that as an update session begins, the local Symmetrix invalidates tracks (16385) on the source (R1) that need updating.

`symrdf -g Rdf2Grp query -i 5`

```
Device Group (DG) Name: Rdf2Grp
DG's Type            : RDF2
DG's Symmetrix ID    : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW   16385        0 RW 009C WD   16385        0 S..   R1 UpdInProg
DEV002  0055 RW   16385        0 RW 009D WD   16385        0 S..   R1 UpdInProg

Total          -------- --------            -------- --------
  Track(s)        32770        0               32770        0
  MB(s)          1024.0      0.0              1024.0      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
```

- As the update progresses, the number of local (R1) invalid tracks as viewed on the source (R1) side keep decreasing because the tracks are being counted down from the original snapshot taken at the beginning of the update process. Meanwhile, the remote (R1) invalid tracks on the target (R2) side continue to be incremented as new writes are executed there.

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI      ST
Standard      A                    N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW   15842        0 RW 009C WD   15125        0 S..   R1 UpdInProg
DEV002  0055 RW   15533        0 RW 009D WD   14891        0 S..   R1 UpdInProg

Total          -------- --------            -------- --------
  Track(s)        31375        0               30016        0
  MB(s)           980.0      0.0               938.0      0.0

Synchronization rate             :     17.2 MB/S
Estimated time to completion     : 00:00:54
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI      ST
Standard      A                    N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW   13040        0 RW 009C WD   12406        0 S..   R1 UpdInProg
DEV002  0055 RW   15819        0 RW 009D WD   12479        0 S..   R1 UpdInProg

Total          -------- --------            -------- --------
  Track(s)        28859        0               24885        0
  MB(s)           901.0      0.0               777.0      0.0

Synchronization rate             :     32.1 MB/S
Estimated time to completion     : 00:00:24
```

- Once the initial 16385 tracks have been updated, the local (R1) invalid tracks on the source (R1) side reach zero, signifying the end of the update operation. During this time, any newly written tracks on the R2 side continue being marked as remote (R1) invalid tracks on the target (R2) side.

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW    9650        0 RW 009C WD       0        0 S..   R1 Updated
DEV002  0055 RW    8574        0 RW 009D WD       0        0 S..   R1 Updated

Total           -------- --------           -------- --------
  Track(s)         18224        0                  0        0
  MB(s)            569.0      0.0                0.0      0.0
```

- To demonstrate the `update -until` option, the example keeps running continuous I/O to the R2 devices and employs two windows: one to provide query displays as the update cycles progress, and one to follow the continuing output from the `symrdf update -until` command.

  In the query window below, the `symrdf query` command from the remote host displays the initial status of the RDF pairs and will redisplay every five seconds. Recall that remote (R1) invalid tracks on the target (R2) side represent continuous I/O to the R2 devices. The local (R1) invalid tracks on the source (R1) side represent the number of tracks that still need to be copied from the target (R2) side (currently zero until the update begins). Only a sampling of the many query displays is shown here, not every one.

`symrdf -g Rdf2Grp query -i 5`

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW   12381        0 RW 009C WD       0        0 S..   R1 Updated
DEV002  0055 RW   12371        0 RW 009D WD       0        0 S..   R1 Updated

Total           -------- --------           -------- --------
  Track(s)         14752        0                  0        0
  MB(s)            459.0      0.0                0.0      0.0
```

- The update window below illustrates the `symrdf update` command from the remote host with the `-until` option track threshold of 100 tracks. While the target (R2) side remains accessible for reads and writes, SYMCLI takes a one-time snapshot of the invalid tracks for each device in the group on the target (R2) side and requests SRDF to begin copying those tracks to the source (R1) side. If SRDF finishes fully copying the snapshot batch of updates to the R1 side and there are still 100 or more R1 (modified) invalid tracks on the target (R2) side, SYMCLI takes another snapshot and requests SRDF to begin copying another batch of tracks to the R1 side. The window displays the series of operations that initiate this first update cycle.

`symrdf -g Rdf2Grp -noprompt update -until 100`

```
An RDF 'Update R1' operation execution is in progress for device group
   'Rdf2Grp'.
   Please wait...

     Suspend RDF link(s).......................................Done.
     Merge device track tables between source and target.......Started.
     Device: 009C ............................................ Merged.
     Device: 009D ............................................ Merged.
     Merge device track tables between source and target.......Done.
     Resume RDF link(s)........................................Done.

The RDF 'Update R1' operation successfully initiated for device group
   'Rdf2Grp'.
```

- The query window below indicates the progression of the first update cycle. As the update progresses, the number of R1 invalid tracks as viewed on the R1 side will continue to decrease as the tracks copied to the R1 device are subtracted from the original snapshot taken at the beginning of the update process. In this update cycle, there are 7379 tracks that remain to be copied from DEV001 on the target (R2) side, and 6767 tracks still to be copied from DEV002 on the target (R2) side. Meanwhile, the R1 (modified) invalid tracks on the R2 side continue to be incremented as new I/O continues there.

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI       ST
Standard      A                    N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW    7658        0 RW 009C WD    7379        0 S..   R1 UpdInProg
DEV002  0055 RW    7026        0 RW 009D WD    6767        0 S..   R1 UpdInProg

Total            -------- --------            -------- --------
  Track(s)          14684        0               14146        0
  MB(s)             458.0      0.0               442.0      0.0

Synchronization rate           :     16.1 MB/S
Estimated time to completion   : 00:00:27
```

- The query window below indicates that the first batch of updates has been fully copied to the R1 side. The local (R1) invalid track count on the R1 side is zero. Because of continuous I/O on the R2 side during the update cycle, the R1 (modified) invalid track count there is 1436 and still over the 100-track threshold. Therefore, SYMCLI will automatically initiate another update cycle. Note, however, that I/O to the R2 side was turned off before the end of this update cycle, which means that this batch of invalid tracks (1436) will be the last batch copied before termination.

```
       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW     757        0 RW 009C WD       0        0 S..   R1 UpdInProg
DEV002  0055 RW     679        0 RW 009D WD       0        0 S..   R1 UpdInProg

Total          -------- --------            -------- --------
  Track(s)         1436        0                   0        0
  MB(s)            44.0      0.0                 0.0      0.0

The update window below indicates the beginning of the second update cycle.
An RDF 'Update R1' operation execution is in progress for device group
   'Rdf2Grp'.
   Please wait...

    Suspend RDF link(s).......................................Done.
    Merge device track tables between source and target.......Started.
    Devices: 009C-009D ...................................... Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s)........................................Done.

The RDF 'Update R1' operation successfully initiated for device group
   'Rdf2Grp'.
```

- The query window below confirms that continuous I/O to the R2 side has stopped. The number of R1 invalid tracks on both the remote site and local site is exactly the same. Recall that when I/O to the R2 side was continuing, the R1 invalid track count there continued to increase and was always greater than the R1 invalid tracks on the R1 side. However, this last batch of updates has not yet been fully copied to the R1 side. Therefore, the RDF pair state still reads R1 UpdInProg.

```
Device Group (DG) Name: Rdf2Grp
DG's Type             : RDF2
DG's Symmetrix ID     : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW     292        0 RW 009C WD     292        0 S..   R1 UpdInProg
DEV002  0055 RW     167        0 RW 009D WD     167        0 S..   R1 UpdInProg
Total          -------- --------            -------- --------
  Track(s)          459        0                 459        0
  MB(s)            14.0      0.0                14.0      0.0
```

- The final query window below shows that the update is complete. The zero count of R1 invalid tracks on the R1 side indicates that this batch was fully copied. The RDF pair state is R1 Updated. The zero count on the total of R1 (modified) invalid tracks on the R2 side indicates a number lower than the 100-track threshold that defined the limit of this update operation.

```
Device Group (DG) Name : Rdf2Grp
DG's Type              : RDF2
DG's Symmetrix ID      : 000000003265

       Target (R2) View                 Source (R1) View     MODES
--------------------------------    ------------------------ ----- ----------
            ST                  LI      ST
Standard     A                   N       A
Logical      T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev  E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0054 RW       0        0 RW 009C WD       0        0 S..   R1 Updated
DEV002  0055 RW       0        0 RW 009D WD       0        0 S..   R1 Updated

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

Example 2: Concurrent RDF

The hardware configuration for the following concurrent RDF example consists of:

- Local Source Symmetrix (sid 77): R1 standard devices 28 and 29
- Remote Target Symmetrix (sid 123): R2 concurrent devices 00 (with 28) and 01 (with 29)
- Remote Target Symmetrix (sid 124): R2 concurrent devices 50 (with 28) and 51 (with 29)

Display outputs may vary slightly according to the version of Solutions Enabler that you are using.

- All commands are issued from the local-site host. The `symcfg list -v` command displays the characteristics of these Symmetrix systems in detail. Note that each Symmetrix system in the configuration must have its "Concurrent RDF Configuration State" set to Enabled, which is a prerequisite for establishing the concurrent SRDF pairs.

`symcfg -v list`

```
Symmetrix ID: 000185400077 (Local)

    Product Model                         : DMX800
    Symmetrix ID                          : 000185400077

    Microcode Version (Number)            : 5670 (16260000)
    Microcode Date                        : 01.05.2004

    Microcode Patch Date                  : 01.05.2004
    Microcode Patch Level                 : 64

    Cache Size                            :    6144 (MB)
    # of Available Cache Slots            :  107946
    Max # of System Write Pending Slots   :   86441
    Max # of DA Write Pending Slots       :   43220
    Max # of Device Write Pending Slots   :    1330

    Symmetrix Total Operating Time        : 62 days, 22:23:35
    Symmetrix Power ON Time               : Tue Nov  4 13:44:30 2003
    Symmetrix Last IPL Time (Cold)        : Mon Dec 15 14:38:35 2003
    Symmetrix Last Fast IPL Time (Hot)    : Mon Jan  5 16:35:51 2004

    Host DB Sync Time                     : Tue Jan  6 11:14:43 2004
    Symmetrix CLI (SYMCLI) Version        : V5.4.0.0 (Edit Level: 516)
    Built with SYMAPI Version             : V5.4.0.0 (Edit Level: 516)
    SYMAPI Run Time Version               : V5.4.0.0 (Edit Level: 516)

    Number of Configured (Sym) Devices    : 938
    Number of Visible (Host) Devices      :   2
    Number of Configured Actual Disks     :  60
    Number of Configured Hot Spares       :   0
    Number of Unconfigured Disks          :   0
    Maximum number of hypers per disk     :  32

    Number of Powerpath Devices           :   2
    Powerpath Run Time Version            : 4.1.0

    SDDF Configuration State              : Enabled
    Configuration Change State            : Enabled
    WORM Configuration Level              : WORM_3
    WORM Characteristics                  : MANUAL_LOCK

    Symmetrix Configuration Checksum      : 2E43F0
    Switched RDF Configuration State      : Disabled
    Concurrent RDF Configuration State    : Enabled
    Dynamic RDF Configuration State       : Enabled
```

```
        Concurrent Dynamic RDF Configuration : Enabled
        RDF Data Mobility Configuration State: Disabled
        Access Control Configuration State   : Enabled
        Device Masking (VCM) Config State    : Disabled
        VCMdb Access Restricted State        : Disabled
        Multi LRU Device Assignment          : BY_NUMBER
        Disk Group Assignments               : Not in Use

        Parity Raid Configuration            : RAID-S (7+1)
        Raid-5 Configuration                 : RAID-5 (7+1)

    Symmetrix ID: 000185400123 (Remote)

        Product Model                        : DMX800
        Symmetrix ID                         : 000185400123

        Microcode Version (Number)           : 5670 (16260000)
        Microcode Date                       : 01.05.2004

        Microcode Patch Date                 : 01.05.2004
        Microcode Patch Level                : 64

        Cache Size                           :    6144 (MB)
        # of Available Cache Slots           :  112671
        # of PermaCache Slots In Use         :    3276
        Max # of System Write Pending Slots  :   90224
        Max # of DA Write Pending Slots      :   45112
        Max # of Device Write Pending Slots  :    1680

        Symmetrix Total Operating Time       : 67 days, 21:40:33
        Symmetrix Power ON Time              : Thu Oct 30 14:27:32 2003
        Symmetrix Last IPL Time (Cold)       : Fri Oct 31 09:21:58 2003
        Symmetrix Last Fast IPL Time (Hot)   : Mon Jan  5 16:32:19 2004

        Host DB Sync Time                    : Tue Jan  6 11:14:43 2004
        Symmetrix CLI (SYMCLI) Version       : V5.4.0.0 (Edit Level: 516)
        Built with SYMAPI Version            : V5.4.0.0 (Edit Level: 516)
        SYMAPI Run Time Version              : V5.4.0.0 (Edit Level: 516)

        Number of Configured (Sym) Devices   : 763
        Number of Visible (Host) Devices     :   0
        Number of Configured Actual Disks    :  60
        Number of Configured Hot Spares      :   0
        Number of Unconfigured Disks         :   0
        Maximum number of hypers per disk    :  32

        Number of Powerpath Devices          :   0
        Powerpath Run Time Version           : 4.1.0

        SDDF Configuration State             : Enabled
        Configuration Change State           : Enabled
        WORM Configuration Level             : WORM_3
        WORM Characteristics                 : MANUAL_LOCK

        Symmetrix Configuration Checksum     : 2622AC
        Switched RDF Configuration State     : Disabled
        Concurrent RDF Configuration State   : Enabled
        Dynamic RDF Configuration State      : Enabled
        Concurrent Dynamic RDF Configuration : Enabled
        RDF Data Mobility Configuration State: Disabled
        Access Control Configuration State   : Enabled
        Device Masking (VCM) Config State    : Disabled
        VCMdb Access Restricted State        : Disabled
        Multi LRU Device Assignment          : BY_NUMBER
        Disk Group Assignments               : Not in Use
```

```
    Parity Raid Configuration             : RAID-S (7+1)
    Raid-5 Configuration                  : RAID-5 (7+1)

Symmetrix ID: 000185400124 (Remote)

    Product Model                         : DMX800
    Symmetrix ID                          : 000185400124

    Microcode Version (Number)            : 5670 (16260000)
    Microcode Date                        : 01.05.2004

    Microcode Patch Date                  : 01.05.2004
    Microcode Patch Level                 : 64

    Cache Size                            :    6144 (MB)
    # of Available Cache Slots            :  112671
    # of PermaCache Slots In Use          :    3276
    Max # of System Write Pending Slots   :   90224
    Max # of DA Write Pending Slots       :   45112
    Max # of Device Write Pending Slots   :    1680

    Symmetrix Total Operating Time        : 67 days, 21:40:33
    Symmetrix Power ON Time               : Thu Oct 30 14:27:32 2003
    Symmetrix Last IPL Time (Cold)        : Fri Oct 31 09:21:58 2003
    Symmetrix Last Fast IPL Time (Hot)    : Mon Jan  5 16:32:19 2004

    Host DB Sync Time                     : Tue Jan  6 11:14:43 2004
    Symmetrix CLI (SYMCLI) Version        : V5.4.0.0 (Edit Level: 516)
    Built with SYMAPI Version             : V5.4.0.0 (Edit Level: 516)
    SYMAPI Run Time Version               : V5.4.0.0 (Edit Level: 516)

    Number of Configured (Sym) Devices    : 763
    Number of Visible (Host) Devices      :   0
    Number of Configured Actual Disks     :  60
    Number of Configured Hot Spares       :   0
    Number of Unconfigured Disks          :   0
    Maximum number of hypers per disk     :  32

    Number of Powerpath Devices           :   0
    Powerpath Run Time Version            : 4.1.0

    SDDF Configuration State              : Enabled
    Configuration Change State            : Enabled
    WORM Configuration Level              : WORM_3
    WORM Characteristics                  : MANUAL_LOCK

    Symmetrix Configuration Checksum      : 2622AC
    Switched RDF Configuration State      : Disabled
    Concurrent RDF Configuration State    : Enabled
    Dynamic RDF Configuration State       : Enabled
    Concurrent Dynamic RDF Configuration : Enabled
    RDF Data Mobility Configuration State: Disabled
    Access Control Configuration State    : Enabled
    Device Masking (VCM) Config State     : Disabled
    VCMdb Access Restricted State         : Disabled
    Multi LRU Device Assignment           : BY_NUMBER
    Disk Group Assignments                : Not in Use

    Parity Raid Configuration             : RAID-S (7+1)
    Raid-5 Configuration                  : RAID-5 (7+1)
```

- The `symrdf list` command with the `-concurrent` option shows devices on the local Symmetrix (sid 77) that are configured as concurrent RDF devices. Note that each of the two concurrent devices of an SRDF concurrent pair belongs to a different RDF group ("RDF Typ:G" 1 and 2). The ellipsis (......) represents truncated output.

`symrdf list -sid 77 -concurrent`

```
Symmetrix ID: 000185400077

                              Local Device View
-------------------------------------------------------------------------
                   STATUS   MODES                     RDF  S T A T E S
Sym         RDF    --------- ----- R1 Inv   R2 Inv ----------------------
Dev  RDev   Typ:G  SA RA LNK  MDA  Tracks   Tracks Dev RDev Pair
---- ----  ------ --------- ----- -------  ------- --- ---- -------------

0028 0050   R1:1  RW RW RW   S..        0        0 RW  WD   Synchronized
     0000   R1:2  RW RW RW   S..        0        0 RW  WD   Synchronized
0029 0051   R1:1  RW RW RW   S..        0        0 RW  WD   Synchronized
     0001   R1:2  RW RW RW   S..        0        0 RW  WD   Synchronized
002A 0052   R1:1  RW RW RW   S..        0        0 RW  WD   Synchronized
     0002   R1:2  RW RW RW   S..        0        0 RW  WD   Synchronized
```

....

- The `sympd list` command displays all Symmetrix devices that are visible to the local host. The display below has been edited to show only those devices that will be used in the example. The N/Grp'd attribute means that these devices are not already part of a device group and are free to be added to a device group.

`sympd list -sid 77`

```
Symmetrix ID: 000185400077

        Device Name           Directors                  Device
--------------------------- ------------ ------------------------------------
                                                                      Cap
Physical                Sym  SA :P DA :IT Config        Attribute    Sts   (MB)
--------------------------- ------------ ------------------------------------
/dev/rdsk/c1t0d32s2      0028 16A:0 02A:C4 RDF1          N/Grp'd      RW     480
/dev/rdsk/c1t0d33s2      0029 16A:0 01B:D1 RDF1          N/Grp'd      RW     480
```

- Creating a device group and adding devices to it are prerequisites for performing SRDF and TimeFinder operations. The `symdg create` command creates a device group named conrdf. The `symld` commands add devices 28 and 29 to it.

```
symdg create conrdf -type rdf1
symld -g conrdf add dev 28
symld -g conrdf add dev 29
```

- The `symrdf query` command displays the status of the SRDF pairs in the device group. The `-rdfg all` option ensures that the display shows the status of both links of a concurrent SRDF pair.

```
symrdf -g conrdf query -rdfg all

Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400124
RDF (RA) Group Number :  1 (00)
Remote Symmetrix ID   : 000185400123
RDF (RA) Group Number :  2 (01)

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW       0        0 RW 0050 WD       0        0 S..   Synchronized
             RW       0        0 RW 0000 WD       0        0 S..   Synchronized
DEV002  0029 RW       0        0 RW 0051 WD       0        0 S..   Synchronized
             RW       0        0 RW 0001 WD       0        0 S..   Synchronized

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

- The `symrdf split` command splits the SRDF pairs in the device group. You can split a concurrent SRDF pair either simultaneously or sequentially. The `-rdfg all` option causes both concurrent devices of an SRDF concurrent pair to be split at the same time.

```
symrdf -g conrdf split -rdfg all -noprompt

An RDF 'Split' operation execution is in progress for device group 'conrdf'.
   Please wait...

    Suspend RDF link(s) ............................................Done.
    Read/Write Enable device(s) in (0077,01) on RA at target (R2)...Done.
    Read/Write Enable device(s) in (0077,02) on RA at target (R2)...Done.

The RDF 'Split' operation successfully executed for device group 'conrdf'.
```

- The `symrdf establish` command performs an incremental establish on the SRDF pairs in the device group. You can establish a concurrent SRDF pair either simultaneously or sequentially. The `-rdfg all` option causes both concurrent devices of an SRDF concurrent pair to be established simultaneously.

`symrdf -g conrdf establish -rdfg all -noprompt`

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

     Write Disable device(s) in (0077,01) on RA at target (R2).......Done.
     Write Disable device(s) in (0077,02) on RA at target (R2).......Done.
     Suspend RDF link(s) for device(s) in (0077,01).................Done.
     Suspend RDF link(s) for device(s) in (0077,02).................Done.
     Mark target device(s) in (0077,01) for incremental copy from
   source..Started.
     Device: 0028 ............................................. Marked.
     Device: 0029 ............................................. Marked.
     Mark target device(s) in (0077,01) for incremental copy from
   source..Done.
     Mark target device(s) in (0077,02) for incremental copy from
   source..Started.
     Device: 0028 ............................................. Marked.
     Device: 0029 ............................................. Marked.
     Mark target device(s) in (0077,02) for incremental copy from
   source..Done.
     Merge track tables between source and target in (0077,01).......Started.
     Device: 0028 ............................................. Merged.
     Device: 0029 ............................................. Merged.
     Merge track tables between source and target in (0077,01).......Done.
     Merge track tables between source and target in (0077,02).......Started.
     Device: 0028 ............................................. Merged.
     Device: 0029 ............................................. Merged.
     Merge track tables between source and target in (0077,02).......Done.
     Resume RDF link(s) for device(s) in (0077,01).................Done.
     Resume RDF link(s) for device(s) in (0077,02).................Done.
The RDF 'Incremental Establish' operation successfully initiated for device
   group 'conrdf'.
```

- The following query shows that the concurrent SRDF pairs are in the process of synchronizing (state is SyncInProg).

symrdf -g conrdf query -rdfg all

```
Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400124
RDF (RA) Group Number :  1 (00)
Remote Symmetrix ID   : 000185400123
RDF (RA) Group Number :  2 (01)

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW    5832        0 RW 0050 WD        0        0 S..   SyncInProg
             RW    5832        0 RW 0000 WD        0        0 S..   SyncInProg
DEV002  0029 RW    8426        0 RW 0051 WD        0        0 S..   SyncInProg
             RW    8426        0 RW 0001 WD        0        0 S..   SyncInProg

Total               -------- --------           -------- --------
  Track(s)           28516        0                  0        0
  MB(s)              891.0      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
```

- The `symrdf verify` command with the `-rdfg all` option displays a message every 30 seconds until both concurrent mirrors of each SRDF pair are synchronized.

symrdf -g conrdf verify -rdfg all -i 30 -synchronized

```
Not all devices in the RDF group 'conrdf' are in the 'Synchronized' state.

Not all devices in the RDF group 'conrdf' are in the 'Synchronized' state.

All devices in the RDF group 'conrdf' are in the 'Synchronized' state.
```

- The `symrdf query` command confirms that both concurrent SRDF pairs are in the Synchronized state.

```
symrdf -g conrdf query -rdfg all

Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400124
RDF (RA) Group Number :  1 (00)
Remote Symmetrix ID   : 000185400123
RDF (RA) Group Number :  2 (01)

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI      ST
Standard      A                    N       A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW       0        0 RW 0050 WD       0        0 S..   Synchronized
             RW       0        0 RW 0000 WD       0        0 S..   Synchronized
DEV002  0029 RW       0        0 RW 0051 WD       0        0 S..   Synchronized
             RW       0        0 RW 0001 WD       0        0 S..   Synchronized

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

- The `symrdf split` command splits the SRDF pairs in the device group. The `-rdfg all` option causes both concurrent devices of an SRDF concurrent pair to be split at the same time.

```
symrdf -g conrdf split -rdfg all -noprompt

An RDF 'Split' operation execution is in progress for device group 'conrdf'.
   Please wait...

    Suspend RDF link(s) ...........................................Done.
    Read/Write Enable device(s) in (0077,01) on RA at target (R2)...Done.
    Read/Write Enable device(s) in (0077,02) on RA at target (R2)...Done.

The RDF 'Split' operation successfully executed for device group 'conrdf'.
```

- If you want to restore data from the concurrent target (R2) devices to their respective source (R1) device, you can restore from one of the concurrent R2 mirrors at any given time. The following `symrdf restore` command with the `-rdfg 2` option causes a restore operation from the concurrent R2 mirror whose link is represented by RDF group 2. (An earlier `symrdf list -concurrent` command displayed which concurrent R2 mirrors belong to which RDF group.)

`symrdf -g conrdf restore -rdfg 2 -noprompt`

```
An RDF 'Incremental Restore' operation execution is in progress for device
   group 'conrdf'. Please wait...

    Write Disable device(s) in (0077,02) on SA at source (R1).......Done.
    Write Disable device(s) in (0077,02) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (0077,02)..................Done.
    Merge track tables between source and target in (0077,02).......Started.
    Devices: 0028-0029 ......................................... Merged.
    Merge track tables between source and target in (0077,02).......Done.
    Resume RDF link(s) for device(s) in (0077,02)...................Done.
    Read/Write Enable device(s) in (0077,02) on SA at source (R1)...Done.

The RDF 'Incremental Restore' operation successfully initiated for device
   group 'conrdf'.
```

- The following query with the `-rdfg 2` option shows the status of each concurrent R2 mirror whose link is represented by RDF group 2. These devices (0000 and 0001) are the concurrent mirrors from which the R1 devices are being restored. The state of the R1s and these R2s is now SyncInProg.

`symrdf -g conrdf query -rdfg 2 -noprompt`

```
Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400123
RDF (RA) Group Number :  2 (01)

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW    5231        0 RW 0000 WD    5122        0 S..   SyncInProg
DEV002  0029 RW    7809        0 RW 0001 WD    7754        0 S..   SyncInProg

Total          -------- --------            -------- --------
  Track(s)        13040        0               12876        0
  MB(s)           407.5      0.0               402.3      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
```

- The following query shows the status of all links of each concurrent SRDF pair. The concurrent R2 mirror from which the restore occurred is now synchronized with its R1 device (state is Synchronized). The other concurrent mirror is still in the Split state.

symrdf -g conrdf query -rdfg all

```
Device Group (DG) Name: conrdf
DG's Type              : RDF1
DG's Symmetrix ID      : 000185400077
Remote Symmetrix ID    : 000185400124
RDF (RA) Group Number : 1 (00)
Remote Symmetrix ID    : 000185400123
RDF (RA) Group Number : 2 (01)

       Source (R1) View                Target (R2) View     MODES
-------------------------------    ------------------------ ----- ----------
            ST                  LI      ST
Standard     A                   N       A
Logical      T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW       0        0 NR 0050 RW       0        0 S..   Split
            RW       0        0 RW 0000 WD       0        0 S..   Synchronized
DEV002  0029 RW       0        0 NR 0051 RW       0        0 S..   Split
            RW       0        0 RW 0001 WD       0        0 S..   Synchronized

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

- If you have written new data to the concurrent R2 mirror that is still in the Split state and you want this data to become the resynchronized data, you can restore again from the split mirror. In this case, however, include the `-remote` option on the `symrdf restore` command line to indicate that you intend to copy data from the split concurrent mirror to both the R1 device and the other (synchronized) concurrent mirror. The `-rdfg 1` option causes the restore operation to occur now from the concurrent mirror whose link is represented by RDF group 1.

symrdf -g conrdf restore -rdfg 1 -remote -noprompt

```
An RDF 'Incremental Restore' operation execution is in progress for device
   group 'conrdf'. Please wait...

    Write Disable device(s) in (0077,01) on SA at source (R1).......Done.
    Write Disable device(s) in (0077,01) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (0077,01)..................Done.
    Merge track tables between source and target in (0077,01).......Started.
    Devices: 0028-0029 ........................................... Merged.
    Merge track tables between source and target in (0077,01).......Done.
    Resume RDF link(s) for device(s) in (0077,01)...................Done.
    Read/Write Enable device(s) in (0077,01) on SA at source (R1)...Done.

The RDF 'Incremental Restore' operation successfully initiated for device
   group 'conrdf'.
```

- The `symrdf verify` command with the `-rdfg 1` option displays a message every 30 seconds until each concurrent R2 mirror represented by RDF group 1 is synchronized with its R1 device.

`symrdf -g conrdf verify -rdfg 1 -i 30 -synchronized`

```
Not all devices in the RDF group 'conrdf' are in the 'Synchronized' state.

Not all devices in the RDF group 'conrdf' are in the 'Synchronized' state.

All devices in the RDF group 'conrdf' are in the 'Synchronized' state.
```

- The `symrdf query` command verifies that both links of the concurrent SRDF pairs are now in the Synchronized state.

`symrdf -g conrdf query -rdfg all`

```
Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400124
RDF (RA) Group Number :  1 (00)
Remote Symmetrix ID   : 000185400123
RDF (RA) Group Number :  2 (01)

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  0028 RW       0        0 RW 0050 WD       0        0 S..   Synchronized
            RW       0        0 RW 0000 WD       0        0 S..   Synchronized
DEV002  0029 RW       0        0 RW 0051 WD       0        0 S..   Synchronized
            RW       0        0 RW 0001 WD       0        0 S..   Synchronized

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

- Currently, in the context of the device group, you can associate a remote BCV with one of the R2 mirrors of a concurrent SRDF pair, but not with both mirrors. Consequently, your device group can include a BCV that belongs to one of the RDF groups, but not both. The following `symbcv associate` command includes in the device group a remotely-associated (`-rdf`) BCV device that belongs to RDF group 1.

`symbcv -g conrdf -rdfg 1 associate dev 14 -rdf`

- The `symmir establish` command fully establishes standard device DEV001 with the remotely-associated BCV. When there are more standard devices in a device group than BCVs, specify which standard device you want to establish.

`symmir -g conrdf -full establish -rdf DEV001 -noprompt -v`

```
Remote 'Full Establish' operation execution is in progress for device
   'DEV001'
   in device group 'conrdf'. Please wait...

PAIRING of Standard and BCV devices:

  Devices: 0050(S) - 0014(B) [PAIRED]

STARTING a BCV 'ESTABLISH' operation.

The BCV 'ESTABLISH' operation SUCCEEDED.

Remote 'Full Establish' operation successfully initiated for device 'DEV001'
   in device group 'conrdf'.
```

- The `symmir query` command with the `-rdf` option shows that RBCV001 (device 14) is now synchronized as a BCV pair with the DEV001 remote R2 mirror (device 50).

`symmir -g conrdf query -rdf -noprompt`

```
Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400124

                     R E M O T E     S Y M M E T R I X

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- ----------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical              Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- ---------

DEV001         0050       0 RBCV001              0014 *          0 Synchronized

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

Legend:

(*): The paired BCV device is associated with this group.
```

- The following `symbcv disassociate` command disassociates BCV device 14 from the device group. The BCV pair remains synchronized even though it is no longer under the control of the device group.

  Note: If you disassociate the BCV at site A from the device group, you can then remotely associate a BCV from site B and create a BCV pair with the concurrent R2 mirror there. However, the BCV pair at site A is no longer under the control of the device group, even though that BCV pair remains synchronized if the pair was in this state when disassociated from the group.

  `symbcv -g conrdf -rdfg 1 disassociate dev 14 -rdf`

- The following `symbcv associate` command includes in the device group a remotely-associated BCV device that belongs to RDF group 2 (that is, this BCV resides on the other remote Symmetrix system).

  `symbcv -g conrdf -rdfg 2 associate dev 61 -rdf`

- The `symmir establish` command fully establishes standard device DEV001 with remotely-associated BCV 61. When there are more standard devices in a device group than BCVs, specify which standard device you want to establish. This BCV is now the only BCV device under the control of the device group.

`symmir -g conrdf -full establish -rdf DEV001 -noprompt -v`

```
Remote 'Full Establish' operation execution is in progress for device
   'DEV001'
   in device group 'conrdf'. Please wait...

PAIRING of Standard and BCV devices:

  Devices: 0000(S) - 0061(B) [PAIRED]

STARTING a BCV 'ESTABLISH' operation.

The BCV 'ESTABLISH' operation SUCCEEDED.

Remote 'Full Establish' operation successfully initiated for device 'DEV001'
   in device group 'conrdf'.
```

- The `symmir query` command with the `-rdf` option shows that RBCV001 (device 0061) is now synchronized as a BCV pair with DEV001's other remote R2 mirror (device 0000).

`symmir -g conrdf query -rdf -noprompt`

```
Device Group (DG) Name: conrdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185400077
Remote Symmetrix ID   : 000185400123

                     R E M O T E     S Y M M E T R I X

     Standard Device                      BCV Device                 State
-------------------------- ------------------------------------- ----------
                    Inv.                                     Inv.
Logical        Sym  Tracks Logical                 Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001         0000      0 RBCV001              0061 *         0 Synchronized

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

Example 3: Creating Dynamic SRDF Pairs

This example is performed using Solutions Enabler version 5.3 and uses the following Symmetrix systems to create dynamic SRDF pairs from non-SRDF devices that are configured for dynamic SRDF capability:

- Local Source Symmetrix (sid 810): RDF-capable standard devices 106, 10A, and 10F
- Remote Target Symmetrix (sid 506): RDF-capable standard devices B7, BF, and C5
- The `symdev list` command with the `-dynamic` option displays devices configured for dynamic RDF capability. When combined with the `-r1` option, `symdev list -dynamic` displays devices configured for dynamic R1/R2[1] and R1-only; when combined with the `-r2` option, the command displays devices configured for dynamic R1/R2 and R2-only. "RDF1+Mir" in the display indicates devices that have already been created as dynamic RDF devices.

  You can use this same command with the `-dynamic` and `-r2` options to list RDF-capable devices on the remote Symmetrix (sid 506) and choose devices there that can be paired as dynamic R2 devices.

```
symdev list -dynamic -sid 810 -r1

Symmetrix ID: 000185500810

       Device Name           Directors                  Device
---------------------------- ------------ ---------------------------------
                                                                       Cap
Sym  Physical                SA :P DA :IT Config        Attribute    Sts   (MB)
---------------------------- ------------ ---------------------------------

00F2 /dev/rdsk/emcpower224c  04A:0 15B:C4 Unprotected   N/Grp'd      RW
  2063
00F5 /dev/rdsk/emcpower227c  04A:0 01A:D1 Unprotected   N/Grp'd      RW
  2063
00FA /dev/rdsk/emcpower239c  04A:0 15A:C4 Unprotected   N/Grp'd      RW
  2063
0106 /dev/rdsk/emcpower235c  04A:0 01B:D3 Unprotected   N/Grp'd      RW
  2063
010A /dev/rdsk/emcpower237c  04A:0 15A:D3 Unprotected   N/Grp'd      RW
  2063
010F /dev/rdsk/emcpower240c  04A:0 02B:C2 Unprotected   N/Grp'd      RW
  2063
0145 Not Visible             04A:0 01A:C1 RDF1+Mir      N/Grp'd      RW    2063
0146 Not Visible             04A:0 15A:C1 RDF1+Mir      N/Grp'd      RW    2063
0147 Not Visible             04A:0 02A:C1 RDF1+Mir      N/Grp'd      RW    2063
0148 Not Visible             04A:0 15A:D1 RDF1+Mir      N/Grp'd      RW    2063
0149 Not Visible             04A:0 02A:D1 3-Way Mir     N/Grp'd      RW    2063
```

- The following command illustrates the use of the vi text editor to create a text file named "devices." As was done here, you can enter into the file those Symmetrix device names that will constitute the dynamic SRDF pairs. The R1 devices are listed in the first column, and the R2 devices (B7, BF, and C5) chosen from the remote Symmetrix are listed in the second column on the same line as their respective R1 source.

```
vi devices
10A  B7
10F  BF
106  C5
```

1. Devices intended for dynamic RDF swap must be configured with the dyn_rdf attribute, which makes a device capable of being either a dynamic R1 device or a dynamic R2 device.

- The `symrdf createpair` command executes the file called "devices" that defines the dynamic SRDF pairs and specifies that the column-1 devices in the file are RDF1 type devices on the local Symmetrix (sid 810). Communication is via RDF group 2. The `-invalidate r2` option invalidates all tracks on the R2 devices in preparation for a subsequent establish operation. The –g option creates a device group named "drdf" and adds the dynamic SRDF pairs to the group.

```
symrdf createpair -file devices -sid 810 -rdfg 2 -invalidate r2 -noprompt
   -type rdf1 -g drdf

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

    Create RDF Pair.........................................Done.
    Mark target device(s) in (0810,02) to refresh from source.......Started.
    Device: 00B7 ..............................................Marked.
    Device: 00BF ..............................................Marked.
    Device: 00C5 ..............................................Marked.
    Mark target device(s) in (0810,02) to refresh from source.......Done.
    Mark target device(s) in (0810,02) for full copy from source....Started.
    Device: 0106 ..............................................Marked.
    Device: 010A ..............................................Marked.
    Device: 010F ..............................................Marked.
    Mark target device(s) in (0810,02) for full copy from source....Done.

The RDF 'Create Pair' operation successfully executed for device file
   'devices'.
```

- The `symrdf query` command shows the status of the dynamic SRDF pairs in the device group (drdf). All three pairs are in the Suspended state.

```
symrdf query -g drdf

Device Group (DG) Name: drdf
DG's Type            : RDF1
DG's Symmetrix ID    : 000185500810

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA    STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  010A WD       0    66000 NR 00B7 WD       0        0 S..   Suspended
DEV002  010F WD       0    66000 NR 00BF WD       0        0 S..   Suspended
DEV003  0106 WD       0    66000 NR 00C5 WD       0        0 S..   Suspended

Total              ------   ------              ------   ------
  Track(s)              0   198000                   0        0
  MB(s)               0.0   6187.0                 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
```

- The `symrdf establish` command initiates copying R1 data to R2 devices. The `-invalidate r2` option from the previous command invalidated the R2 devices, a step that is usually carried out during a full establish operation. Consequently, you do not need the `-full` option here. The invalidate step is not repeated, regardless of whether you use the `-full` option or not. If subsequently you re-establish or restore the dynamic SRDF pairs, omitting or including the `-full` option will affect how the copy occurs (either incremental copy or full copy, respectively). The output below says "Incremental Establish" because the `-full` option was omitted. However, because all tracks on the R2 devices were previously invalidated, the result is a full copy of all R1 tracks to the R2 tracks.

`symrdf establish -g drdf -noprompt`

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

     Suspend RDF link(s).......................................Done.
     Read/Write Enable device(s) on SA at source (R1)..........Done.
     Resume RDF link(s)........................................Done.

The RDF 'Incremental Establish' operation successfully initiated for device
   group 'drdf'.
```

- The following query displays the status of the dynamic SRDF pairs. The pairs are currently in the process of synchronizing (pair state is SyncInProg).

`symrdf query -g drdf`

```
Device Group (DG) Name: drdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185500810

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI      ST
Standard      A                    N       A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  010A RW       0    59491 RW 00B7 WD       0        0 S..   SyncInProg
DEV002  010F RW       0    61732 RW 00BF WD       0        0 S..   SyncInProg
DEV003  0106 RW       0    64059 RW 00C5 WD       0        0 S..   SyncInProg

Total            ------   ------             ------   ------
  Track(s)            0   185282                  0        0
  MB(s)             0.0   5782.7                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
```

- The `symrdf verify` command verifies when the dynamic SRDF pairs have reached the Synchronized state. The ellipsis (……) represents repetitive output that was omitted.

`symrdf verify -g drdf -i 5 -synchronized`

```
NONE of the mirrored pairs are in the 'Synchronized' state

NONE of the mirrored pairs are in the 'Synchronized' state
.........................................................................................................................

All devices in the RDF group 'drdf' are in the 'Synchronized' stateAnother
   query confirms that the SRDF pairs are now in the Synchronized state.
```

`symrdf query -g drdf`

```
Device Group (DG) Name: drdf
DG's Type             : RDF1
DG's Symmetrix ID     : 000185500810

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  010A RW       0        0 RW 00B7 WD       0        0   S.. Synchronized
DEV002  010F RW       0        0 RW 00BF WD       0        0   S.. Synchronized
DEV003  0106 RW       0        0 RW 00C5 WD       0        0   S.. Synchronized

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

- The `symrdf split` command splits all dynamic SRDF pairs in the device group.

`symrdf split -g drdf -noprompt`

```
An RDF 'Split' operation execution is in progress for device group 'drdf'.
   Please wait...

    Suspend RDF link(s).......................................Done.
    Read/Write Enable device(s) on RA at target (R2)..........Done.

The RDF 'Split' operation successfully executed for device group 'drdf'.
```

- The symrdf query command confirms that the SRDF pairs are in the Split state.

```
symrdf query -g drdf

Device Group (DG) Name: drdf
DG's Type              : RDF1
DG's Symmetrix ID      : 000185500810

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ---------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  010A RW       0        0 NR 00B7 RW       0        0 S..   Split
DEV002  010F RW       0        0 NR 00BF RW       0        0 S..   Split
DEV003  0106 RW       0        0 NR 00C5 RW       0        0 S..   Split

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

- Once the pairs' link state is Not Ready (NR), you can use the symrdf deletepair command to cancel the dynamic SRDF pairings defined in the device group and delete this pairing information from the host's SYMAPI database file. This operation also changes the type of the device group from RDF1 to REGULAR; the devices in the device group are changed from R1 devices to standard devices.

```
symrdf deletepair -g drdf -noprompt

An RDF 'Delete Pair' operation execution is in progress for device group
   'drdf'.
   Please wait...

     Delete RDF Pair...........................................Done.

The RDF 'Delete Pair' operation successfully executed for device group
   'drdf'.
```

- Attempting to perform a symrdf query on the device group results in an output confirming that this device group is no longer an RDF1 type group as a result of the symrdf deletepair command.

```
symrdf query -g drdf

Device group 'drdf' is not an RDF group.
```

- The `symld list` command on the device group shows that the device group type was changed to REGULAR and that the same devices that had been created as dynamic R1 type devices have returned to being standard devices (although still configured as RDF-capable). These devices lost their SRDF characteristics as a result of the `symrdf deletepair` command.

`symld list -g drdf`

```
Device Group (DG) Name: drdf
DG's Type             : REGULAR
DG's Symmetrix ID     : 000185500810

       Standard Device Name           Directors             Device
-----------------------------------   ------------ -------------------------
                                                                      Cap
Logical      Physical           Sym    SA :P DA :IT Config       Att Sts   (MB)
-----------------------------------   ------------ -------------------------

DEV001      emcpower237c       010A   04A:0 15A:D3 Unprotected       WD    2063
DEV002      emcpower240c       010F   04A:0 02B:C2 Unprotected       WD    2063
DEV003      emcpower235c       0106   04A:0 01B:D3 Unprotected       WD    2063
```

Example 4: Creating a Dynamic RDF Group

The hardware setup consists of two Symmetrix arrays (sid 6190 and sid 0257) that are connected to each other and to two remote Symmetrix arrays (sid 6202 and sid 0254).

- The `symcfg list` command displays those Symmetrix arrays that are visible to this host. Note that two Symmetrix arrays are running Enginuity Version 5568, and two are running version 5669. Creating a dynamic RDF group is possible only for Symmetrix arrays running version 5669 or higher.

`symcfg list`

```
                              S Y M M E T R I X

                                       Mcode     Cache      Num Phys  Num Symm
    SymmID       Attachment  Model     Version   Size (MB)  Devices   Devices

    000000006190 Local       DMX2000P  5669        20480      100        396
    000184600257 Local       8230      5568        16384       79        483
    000000006202 Remote      DMX2000P  5669        20480        0        534
    000184600254 Remote      8230      5568        16384        0        504
```

- The `symcfg list -ra all` command displays the RDF (RA) groups of all connected Symmetrix arrays (one or two hops away) that are accessible through RDF links. The `-switched` option displays whether the RDF group type is dynamic (Enginuity Version 5669 or higher) or static. If you query a Symmetrix running Enginuity Version 5569 or higher, a group's label name is displayed (the default is RDFDVGROUP). Symmetrix 6190 has multiple links to remote Symmetrix 6202 and a single link to local Symmetrix 0257.

`symcfg list -ra all -switched`

```
Symmetrix ID: 000000006190 (Local)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local               Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb  RA Grp  Type    Name        SymmID        Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------  ------ ---- -------

    RF-14A 14A   60 (3B) Dynamic DYNGRP60    000000006202  RF-14A 14A   60 (3B)
                 55 (36) Static  DYNGRP55    000000006202  RF-14A 14A    4 (03)
                  1 (00) Dynamic DYNGRP1     000000006202  RF-14A 14A    1 (00)

    RF-14B 14B    8 (07) Static  HOUSTON     000000006202  RF-14C 14C    9 (08)
                 21 (14) Static  RDFDVGROUP  000000006202  RF-14C 14C   18 (11)

    RF-14C 14C    5 (04) Static  RDFDVGROUP  000000006202  RF-14D 14D   49 (30)

    RF-14D 14D    6 (05) Static  HOPKINTON   000184600257  RF-16A 16A    4 (D)

Symmetrix ID: 000184600257 (Local)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local               Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb  RA Grp  Type    Name        SymmID        Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------  ------ ---- -------

    RF-3A  03A    2 (B)  Static  -           000000006202  RF-14A 14A    3 (02)

    RF-16A 16A    4 (D)  Static  -           000000006190  RF-14D 14D    6 (05)
```

```
    RF-3B   03B    5 (E)    Static  -          000184600254 RF-16B 16B    8 (H)
                   7 (G)    Static  -          000184600254 RF-16B 16B    3 (C)

    RF-16B 16B    -        Static  -          -            -      -      -

Symmetrix ID: 000000006202 (Remote)

            S Y M M E T R I X    R D F    D I R E C T O R S

        Local               Group                     Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-14A 14A   60 (3B)  Dynamic DYNGRP60    000000006190 RF-14A 14A   60 (3B)
                  3 (02)  Static  RDFDVGROUP  000184600257 RF-3A  03A    2 (B)
                  4 (03)  Static  RDFDVGROUP  000000006190 RF-14A 14A   55 (36)
                  1 (00)  Dynamic DYNGRP1     000000006190 RF-14A 14A    1 (00)

    RF-14B 14B   10 (09)  Static  RDFDVGROUP  000184600254 RF-16A 16A    2 (B)
                  5 (04)  Static  RDFDVGROUP  000184600254 RF-16A 16A   10 (J)

    RF-14C 14C    9 (08)  Static  RDFDVGROUP  000000006190 RF-14B 14B    8 (07)
                 18 (11)  Static  RDFDVGROUP  000000006190 RF-14B 14B   21 (14)

    RF-14D 14D   49 (30)  Static  RDFDVGROUP  000000006190 RF-14C 14C    5 (04)

Symmetrix ID: 000184600254 (Remote)

            S Y M M E T R I X    R D F    D I R E C T O R S

        Local               Group                     Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-16A 16A    2 (B)   Static  -           000000006202 RF-14B 14B   10 (09)
                 10 (J)   Static  -           000000006202 RF-14B 14B    5 (04)

    RF-16B 16B    8 (H)   Static  -           000184600257 RF-3B  03B    5 (E)
                  3 (C)   Static  -           000184600257 RF-3B  03B    7 (G)
```

- The following `symrdf addgrp` command creates a dynamic RDF group that represents another RDF link between Symmetrix 6190 and Symmetrix 6202. It adds dynamic RDF group 63 on the local Symmetrix 6190, and RDF group 63 on the remote Symmetrix 6202. The command requires that you specify a group label (DYNGRP63 in this case) that can be used when modifying or deleting the group. Creation of the local and remote RDF groups includes director 14A from both the local and remote Symmetrix array. It is not necessary that the RDF group number or the director on the local and remote Symmetrix arrays be the same.

  However, it is important to be aware of your network topology when creating dynamic RDF groups between two Symmetrix arrays. To create a dynamic RDF link (a connection) between directors, the director end points must be able to see each other through the Fibre Channel fabric. For example, a dynamic RDF link can be created between director 14A on Symmetrix 6190 and director 14D on Symmetrix 6202 only if the Fibre Channel zoning is set up so that the two directors can see each other through the fabric.

  ```
  symrdf addgrp -label DYNGRP63 -rdfg 63 -sid 90 -dir 14a \
     -remote_rdfg 63 -remote_sid 02 -remote_dir 14a

   Successfully Added Dynamic RDF Group 'DYNGRP63' for Symm: 000000006190
  ```

- Another `symcfg list -ra all` command with the `-switched` option verifies that RDF group 63 (DYNGRP63) has been added to both the local and remote Symmetrix arrays. A Symmetrix array running Enginuity Version 5669 or higher can have up to 64 RDF groups, each group having its hexadecimal value[1] in parenthesis. Symmetrix 0257 and 0254, which are running Enginuity Version 5568, always display Group Type as "Static" because dynamic RDF groups are valid only for version 5669 and higher.

`symcfg list -ra all -switched`

```
Symmetrix ID: 000000006190 (Local)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local                Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-14A 14A  60 (3B)  Dynamic DYNGRP60    000000006202 RF-14A 14A  60 (3B)
                55 (36)  Static  DYNGRP55    000000006202 RF-14A 14A   4 (03)
                 1 (00)  Dynamic DYNGRP1     000000006202 RF-14A 14A   1 (00)
                63 (3E)  Dynamic DYNGRP63    000000006202 RF-14A 14A  63 (3E)

    RF-14B 14B   8 (07)  Static  HOUSTON     000000006202 RF-14C 14C   9 (08)
                21 (14)  Static  RDFDVGROUP  000000006202 RF-14C 14C  18 (11)

    RF-14C 14C   5 (04)  Static  RDFDVGROUP  000000006202 RF-14D 14D  49 (30)

    RF-14D 14D   6 (05)  Static  HOPKINTON   000184600257 RF-16A 16A   4 (D)

Symmetrix ID: 000184600257 (Local)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local                Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-3A  03A   2 (B)   Static  -           000000006202 RF-14A 14A   3 (02)

    RF-16A 16A   4 (D)   Static  -           000000006190 RF-14D 14D   6 (05)

    RF-3B  03B   5 (E)   Static  -           000184600254 RF-16B 16B   8 (H)
                 7 (G)   Static  -           000184600254 RF-16B 16B   3 (C)

    RF-16B 16B   -       Static  -           -            -      -    -

Symmetrix ID: 000000006202 (Remote)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local                Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-14A 14A  60 (3B)  Dynamic DYNGRP60    000000006190 RF-14A 14A  60 (3B)
                 3 (02)  Static  RDFDVGROUP  000184600257 RF-3A  03A   2 (B)
```

1. Prior to Enginuity Version 5669, the maximum number of RDF groups was 16, and groups created under those versions are displayed as letters A through P. For version 5669 and higher, the maximum number is 64, and each group is displayed as a hex value that is one less than its decimal value (internal to the Symmetrix, RDF groups are 0-based; from the SYMCLI point of view, they are 1-based).

```
                   4 (03)   Static  RDFDVGROUP  000000006190 RF-14A 14A  55 (36)
                   1 (00)   Dynamic DYNGRP1     000000006190 RF-14A 14A   1 (00)
                  63 (3E)   Dynamic DYNGRP63    000000006190 RF-14A 14A  63 (3E)

    RF-14B 14B    10 (09)   Static  RDFDVGROUP  000184600254 RF-16A 16A   2 (B)
                   5 (04)   Static  RDFDVGROUP  000184600254 RF-16A 16A  10 (J)

    RF-14C 14C     9 (08)   Static  RDFDVGROUP  000000006190 RF-14B 14B   8 (07)
                  18 (11)   Static  RDFDVGROUP  000000006190 RF-14B 14B  21 (14)

    RF-14D 14D    49 (30)   Static  RDFDVGROUP  000000006190 RF-14C 14C   5 (04)

Symmetrix ID: 000184600254 (Remote)

            S Y M M E T R I X    R D F    D I R E C T O R S

           Local                Group                      Remote
    -------------------  ------------------  --------------------------------
    Ident  Symb RA Grp   Type    Name        SymmID       Ident  Symb RA Grp
    ------ ---- -------  ------- ----------  ------------ ------ ---- -------

    RF-16A 16A   2 (B)   Static  -           000000006202 RF-14B 14B  10 (09)
                10 (J)   Static  -           000000006202 RF-14B 14B   5 (04)

    RF-16B 16B   8 (H)   Static  -           000184600257 RF-3B  03B   5 (E)
                 3 (C)   Static  -           000184600257 RF-3B  03B   7 (G)
```

Example 5: Operating with SRDF Asynchronous Replication

This example is performed using Solutions Enabler version 5.3. The hardware setup consists of a host connected to a source Symmetrix (sid 6163) running Enginuity Version 5670 and remotely connected via RDF links to a target Symmetrix (sid 6201) that is also running version 5670. RDF (RA) group number 3 has been configured to provide SRDF/A operations.

- The `symrdf list` command with the `-rdfa` option displays all devices that are configured for SRDF/A operation. (Beginning with Solutions Enabler Version 6.0, all RDF groups on a Symmetrix array are capable of SRDF/A operation; prior versions required that you configure an RDF group to be capable of SRDF/A.) The "G" column indicates that RDF group number 3 is the SRDF/A-configured group. Devices in this type of RDF group have to be either all R1 devices or all R2 devices.

`symrdf list -rdfa`

```
Symmetrix ID: 000000006163

                              Local Device View
-------------------------------------------------------------------------
                  STATUS    MODES                          RDF  S T A T E S
Sym        RDF    --------- -----  R1 Inv    R2 Inv ----------------------
Dev  RDev  Typ:G  SA RA LNK  MDA   Tracks    Tracks Dev RDev Pair
---- ---- ------ --------- ----- -------  ------- --- ---- -------------
00F2 00E6  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F3 00E7  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F4 00E8  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F5 00E9  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F6 00EA  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F7 00EB  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F8 00EC  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00F9 00ED  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FA 00EE  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FB 00EF  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FC 0104  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FD 0105  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FE 0106  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
00FF 0107  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
0100 0108  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized
0101 0109  R1:3  RW RW RW   S..        0         0 RW  WD   Synchronized

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

- The `symdg` command creates an RDF1 type device group named AsyncGrp1. The `symld` command adds all devices from RDF group 3 to the device group. All devices in RDF group 3 must be managed together using async replication; no subset of this group can be managed using async replication.

```
symdg create AsyncGrp1 -type rdf1
symld -g AsyncGrp1 addall -rdfg 3
```

- The `symrdf query` command displays the status of the SRDF pairs in the device group. Currently the pairs are in the Synchronized state and running with Synchronous (S) replication. As is shown later in this example, you can include the `-rdfa` option to display SRDF/A information such as the session number, cycle number, and session status (which is currently inactive).

`symrdf -g AsyncGrp1 query`

```
Device Group (DG) Name              : AsyncGrp1
DG's Type                           : RDF1
DG's Symmetrix ID                   : 000000006163

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------
DEV001  00F2 RW       0        0 RW 00E6  WD       0        0 S..   Synchronized
DEV002  00F3 RW       0        0 RW 00E7  WD       0        0 S..   Synchronized
DEV003  00F4 RW       0        0 RW 00E8  WD       0        0 S..   Synchronized
DEV004  00F5 RW       0        0 RW 00E9  WD       0        0 S..   Synchronized
DEV005  00F6 RW       0        0 RW 00EA  WD       0        0 S..   Synchronized
DEV006  00F7 RW       0        0 RW 00EB  WD       0        0 S..   Synchronized
DEV007  00F8 RW       0        0 RW 00EC  WD       0        0 S..   Synchronized
DEV008  00F9 RW       0        0 RW 00ED  WD       0        0 S..   Synchronized
DEV009  00FA RW       0        0 RW 00EE  WD       0        0 S..   Synchronized
DEV010  00FB RW       0        0 RW 00EF  WD       0        0 S..   Synchronized
DEV011  00FC RW       0        0 RW 0104  WD       0        0 S..   Synchronized
DEV012  00FD RW       0        0 RW 0105  WD       0        0 S..   Synchronized
DEV013  00FE RW       0        0 RW 0106  WD       0        0 S..   Synchronized
DEV014  00FF RW       0        0 RW 0107  WD       0        0 S..   Synchronized
DEV015  0100 RW       0        0 RW 0108  WD       0        0 S..   Synchronized
DEV016  0101 RW       0        0 RW 0109  WD       0        0 S..   Synchronized

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

- The `symrdf set mode async` command sets the method of replication to Asynchronous for the SRDF/A devices in the device group. At this point, however, consistency protection is still disabled.

`symrdf -g AsyncGrp1 set mode async -noprompt`

```
An RDF Set 'Asynchronous Mode' operation execution is in progress for device
   group 'AsyncGrp1'. Please wait...

The RDF Set 'Asynchronous Mode' operation successfully executed for device
   group 'AsyncGrp1'.
```

- The `symrdf enable` command enables consistency protection for the SRDF/A devices in the device group.

```
symrdf -g AsyncGrp1 enable -noprompt

An RDF 'Enable' operation execution is in progress for device group
   'AsyncGrp1'.
   Please wait...

The RDF 'Enable' operation successfully executed for device group
   'AsyncGrp1'.
```

- The `symdg show` display verifies in its "RDFA Information" section that SRDF/A session is active and that the consistency state is enabled.

```
symdg show AsyncGrp1

Group Name:  AsyncGrp1

    Group Type                                   : RDF1     (RDFA)
    Valid                                        : Yes
    Symmetrix ID                                 : 000000006163
    Group Creation Time                          : Mon Jun 30 14:02:12 2003
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :   16
    Number of Associated GK's                    :    0
    Number of Locally-associated BCV's           :    0
    Number of Locally-associated VDEV's          :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (16):
        {
        --------------------------------------------------------------------
                                                       Sym                Cap
        LdevName              PdevName                 Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/emcpower99c    00F2      RW      1031
        DEV002                /dev/rdsk/emcpower100c   00F3      RW      1031
        DEV003                /dev/rdsk/emcpower101c   00F4      RW      1031
        DEV004                /dev/rdsk/emcpower102c   00F5      RW      1031
        DEV005                /dev/rdsk/emcpower103c   00F6      RW      1031
        DEV006                /dev/rdsk/emcpower104c   00F7      RW      1031
        DEV007                /dev/rdsk/emcpower105c   00F8      RW      1031
        DEV008                /dev/rdsk/emcpower106c   00F9      RW      1031
        DEV009                /dev/rdsk/emcpower107c   00FA      RW      1031
        DEV010                /dev/rdsk/emcpower108c   00FB      RW      1031
        DEV011                /dev/rdsk/emcpower109c   00FC      RW      1031
        DEV012                /dev/rdsk/emcpower110c   00FD      RW      1031
        DEV013                /dev/rdsk/emcpower111c   00FE      RW      1031
        DEV014                /dev/rdsk/emcpower112c   00FF      RW      1031
        DEV015                /dev/rdsk/emcpower113c   0100      RW      1031
        DEV016                /dev/rdsk/emcpower114c   0101      RW      1031
        }

    Device Group RDF Information
        {
        RDF Type                               : R1
        RDF (RA) Group Number                  : 3               (02)

        Remote Symmetrix ID                    : 000000006201

        R2 Device Is Larger Than The R1 Device : False
```

```
RDF Mode                                  : Asynchronous
RDF Adaptive Copy                         : Disabled
RDF Adaptive Copy Write Pending State     : N/A
RDF Adaptive Copy Skew (Tracks)           : 65535

RDF Device Domino                         : Disabled

RDF Link Configuration                    : Fibre
RDF Link Domino                           : Disabled
Prevent Automatic RDF Link Recovery       : Disabled
Prevent RAs Online Upon Power ON          : Enabled

Device RDF Status                         : Ready           (RW)

Device RA Status                          : Ready           (RW)
Device Link Status                        : Ready           (RW)

Device Suspend State                      : N/A
Device Consistency State                  : Disabled
RDF R2 Not Ready If Invalid               : Enabled

Device RDF State                          : Ready           (RW)
Remote Device RDF State                   : Write Disabled  (WD)

RDF Pair State (  R1 <===> R2 )           : Consistent

Number of R1 Invalid Tracks               : 0
Number of R2 Invalid Tracks               : 0

RDFA Information:
    {
    Session Number                        : 0
    Cycle Number                          : 5
    Number of Devices in the Session      : 16
    Session Status                        : Active

    Session Consistency State             : Enabled
    Tracks not Committed to the R2 Side: 0
    Average Cycle Time                    : 00:00:30
    Time that R2 is behind R1             : 00:00:46
    }
}
```

- In the RDFA information display above, "Tracks not Committed to the R2 Side" indicates all R1 tracks owed to the R2 side that have not been committed to the R2 side yet. The "Average Cycle Time" is 30 seconds. The "Time that R2 is behind R1" indicates that data on the R2 side is currently 46 seconds behind the R1 side.

  The `symrdf verify` command checks the state of the SRDF pairs and verifies that they are in the Consistent state.

  `symrdf -g AsyncGrp1 verify -consistent`

  ```
  All devices in the RDF group 'AsyncGrp1' are in the 'Consistent' state.
  ```

- The `symrdf suspend` command with the `-force` option trips the device group, making the devices NR on the link and disabling SRDF/A consistency protection. Suspending is useful if you need to trip the device group but also maintain the consistency of the R2 database copy with the production copy on the R1 side. The `-force` option is required here to ensure that you really want to stop SRDF/A operation and end consistency protection.

`symrdf -g AsyncGrp1 suspend -noprompt -force`

```
An RDF 'Suspend' operation execution is in progress for device group
   'AsyncGrp1'.
   Please wait...

     Suspend RDF link(s).......................................Done.

The RDF 'Suspend' operation successfully executed for device group
   'AsyncGrp1'.
```

- The `symrdf query` command with the `-rdfa` option shows that the SRDF/A session status is now inactive and that the SRDF pairs are in the Suspended state. Normally there would be invalid tracks on the R1 side to indicate continuing I/O on the R1 side, but currently there is no I/O.

`symrdf -g AsyncGrp1 query -rdfa`

```
Device Group (DG) Name                  : AsyncGrp1
DG's Type                               : RDF1
DG's Symmetrix ID                       : 000000006163
RDFA Session Number                     : 0
RDFA Cycle Number                       : 0
RDFA Session Status                     : Inactive
RDFA Avg Cycle Time                     : 00:00:00
Time that R2 is behind R1               : 00:00:00

       Source (R1) View                    Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ---------
DEV001  00F2 RW       0        0 NR 00E6 WD       0        0 A..-  Suspended
DEV002  00F3 RW       0        0 NR 00E7 WD       0        0 A..-  Suspended
DEV003  00F4 RW       0        0 NR 00E8 WD       0        0 A..-  Suspended
DEV004  00F5 RW       0        0 NR 00E9 WD       0        0 A..-  Suspended
DEV005  00F6 RW       0        0 NR 00EA WD       0        0 A..-  Suspended
DEV006  00F7 RW       0        0 NR 00EB WD       0        0 A..-  Suspended
DEV007  00F8 RW       0        0 NR 00EC WD       0        0 A..-  Suspended
DEV008  00F9 RW       0        0 NR 00ED WD       0        0 A..-  Suspended
DEV009  00FA RW       0        0 NR 00EE WD       0        0 A..-  Suspended
DEV010  00FB RW       0        0 NR 00EF WD       0        0 A..-  Suspended
DEV011  00FC RW       0        0 NR 0104 WD       0        0 A..-  Suspended
DEV012  00FD RW       0        0 NR 0105 WD       0        0 A..-  Suspended
DEV013  00FE RW       0        0 NR 0106 WD       0        0 A..-  Suspended
DEV014  00FF RW       0        0 NR 0107 WD       0        0 A..-  Suspended
DEV015  0100 RW       0        0 NR 0108 WD       0        0 A..-  Suspended
DEV016  0101 RW       0        0 NR 0109 WD       0        0 A..-  Suspended

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

- The `symrdf resume` command resumes the RDF links between the SRDF pairs in the device group and I/O traffic between the R1 devices and their paired R2 devices. The SRDF/A session is automatically activated again.

```
symrdf -g AsyncGrp1 resume -noprompt

An RDF 'Resume' operation execution is in progress for device group
   'AsyncGrp1'.
   Please wait...

     Resume RDF link(s)........................................Done.

The RDF 'Resume' operation successfully executed for device group
   'AsyncGrp1'.
```

- At this point, the SRDF/A devices are ready again on the RDF link and operating with Asynchronous replication. Consistency protection remains set The `symrdf query` command verifies that the SRDF/A session is active again and that the devices are in the Consistent state.

```
symrdf -g AsyncGrp1 query -rdfa

Device Group (DG) Name                : AsyncGrp1
DG's Type                             : RDF1
DG's Symmetrix ID                     : 000000006163
RDFA Session Number                   : 0
RDFA Cycle Number                     : 6
RDFA Session Status                   : Active
RDFA Avg Cycle Time                   : 00:00:30
Time that R2 is behind R1             : 00:00:38

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ---------
DEV001  00F2 RW       0        0 RW 00E6 WD       0        0 A..X  Consistent
DEV002  00F3 RW       0        0 RW 00E7 WD       0        0 A..X  Consistent
DEV003  00F4 RW       0        0 RW 00E8 WD       0        0 A..X  Consistent
DEV004  00F5 RW       0        0 RW 00E9 WD       0        0 A..X  Consistent
DEV005  00F6 RW       0        0 RW 00EA WD       0        0 A..X  Consistent
DEV006  00F7 RW       0        0 RW 00EB WD       0        0 A..X  Consistent
DEV007  00F8 RW       0        0 RW 00EC WD       0        0 A..X  Consistent
DEV008  00F9 RW       0        0 RW 00ED WD       0        0 A..X  Consistent
DEV009  00FA RW       0        0 RW 00EE WD       0        0 A..X  Consistent
DEV010  00FB RW       0        0 RW 00EF WD       0        0 A..X  Consistent
DEV011  00FC RW       0        0 RW 0104 WD       0        0 A..X  Consistent
DEV012  00FD RW       0        0 RW 0105 WD       0        0 A..X  Consistent
DEV013  00FE RW       0        0 RW 0106 WD       0        0 A..X  Consistent
DEV014  00FF RW       0        0 RW 0107 WD       0        0 A..X  Consistent
DEV015  0100 RW       0        0 RW 0108 WD       0        0 A..X  Consistent
DEV016  0101 RW       0        0 RW 0109 WD       0        0 A..X  Consistent

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

Example 6: Using a Composite Group to Contol SRDF Pairs

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host connected to two source Symmetrix arrays (Symmetrix 000187900035 and Symmetrix 000000003143). The example builds a composite group with source R1 devices from both Symmetrix arrays. It then demonstrates how to enable consistency protection for the composite group and perform a suspend operation on the group. For more examples using SRDF consistency protection, refer to *Operational Examples, Chapters 8* and *9*.

- The `symcg create` command creates an RDF1 type composite group named SRDF on this host. If you intend to enable the group for consistency protection and have not set the `SYMAPI_RDF_CG_TO_PPATH` variable to `ENABLE`, you must include the `-ppath` option so that the group is added to PowerPath.

  `symcg create SRDF -type rdf1 -ppath`

- The following `symcg addall` command adds to the composite group a range of PowerPath standard devices from Symmetrix 000187900035.

  `symcg -cg SRDF addall dev -range 137:14F -sid 35`

- The following `symcg addall` command adds to the composite group a range of PowerPath standard devices from Symmetrix 000000003143.

  `symcg -cg SRDF addall dev -range F7:10F -sid 43`

- The `symrdf query` command checks the state of the SRDF pairs. Note that SRDF pairs from one Symmetrix array are in the Suspended state, while the other Symmetrix array has synchronized SRDF pairs.

`symrdf -cg SRDF query`

```
Composite Group Name     :  SRDF
Composite Group Type     :  RDF1
Number of Symmetrix Arrays :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number       :  1 (A)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- -----  ------ ------------
     ST                 LI       ST                       C   S
      A                  N        A                       o   u
      T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       n   s   RDF Pair
Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDA   s   p   STATE
-----------------------  -- ---------------------- -----  ------ ------------
00F7 RW       0      46 NR 0062 NR       0       0 S..    .  -   Suspended
00FA RW       0      46 NR 0065 NR       0       0 S..    .  -   Suspended
00FC RW       0      46 NR 0067 NR       0       0 S..    .  -   Suspended
00FD RW       0      46 NR 0068 NR       0       0 S..    .  -   Suspended
00FE RW       0      46 NR 0069 NR       0       0 S..    .  -   Suspended
00FF RW       0      46 NR 006A NR       0       0 S..    .  -   Suspended
0100 RW       0      46 NR 006B NR       0       0 S..    .  -   Suspended
0101 RW       0      46 NR 006C NR       0       0 S..    .  -   Suspended
0102 RW       0      46 NR 006D NR       0       0 S..    .  -   Suspended
0103 RW       0      46 NR 006E NR       0       0 S..    .  -   Suspended
0104 RW       0      46 NR 006F NR       0       0 S..    .  -   Suspended
0105 RW       0      46 NR 0070 NR       0       0 S..    .  -   Suspended
0106 RW       0      46 NR 0071 NR       0       0 S..    .  -   Suspended
0107 RW       0      46 NR 0072 NR       0       0 S..    .  -   Suspended
0108 RW       0      46 NR 0073 NR       0       0 S..    .  -   Suspended
0109 RW       0      46 NR 0074 NR       0       0 S..    .  -   Suspended
```

```
010A RW        0       46 NR 0075 NR        0        0 S..    .  -   Suspended
010B RW        0       46 NR 0076 NR        0        0 S..    .  -   Suspended
010C RW        0       46 NR 0077 NR        0        0 S..    .  -   Suspended
010D RW        0       46 NR 0078 NR        0        0 S..    .  -   Suspended
010E RW        0       46 NR 0079 NR        0        0 S..    .  -   Suspended
010F RW        0       46 NR 007A NR        0        0 S..    .  -   Suspended
Symmetrix ID                  : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID           : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number         :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
----------------------- ------------------------- ----- ------ ------------
     ST                LI       ST                      C  S
      A                 N        A                      o  u
      T  R1 Inv  R2 Inv K        T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
0137 RW        0        0 RW 0056 NR        0        0 S..    .  -   Synchronized
013A RW        0        0 RW 0059 NR        0        0 S..    .  -   Synchronized
013C RW        0        0 RW 005B NR        0        0 S..    .  -   Synchronized
013D RW        0        0 RW 005C NR        0        0 S..    .  -   Synchronized
013E RW        0        0 RW 005D NR        0        0 S..    .  -   Synchronized
013F RW        0        0 RW 005E NR        0        0 S..    .  -   Synchronized
0140 RW        0        0 RW 005F NR        0        0 S..    .  -   Synchronized
0141 RW        0        0 RW 0060 NR        0        0 S..    .  -   Synchronized
0142 RW        0        0 RW 0061 NR        0        0 S..    .  -   Synchronized
0143 RW        0        0 RW 0062 NR        0        0 S..    .  -   Synchronized
0144 RW        0        0 RW 0063 NR        0        0 S..    .  -   Synchronized
0145 RW        0        0 RW 0064 NR        0        0 S..    .  -   Synchronized
0146 RW        0        0 RW 0065 NR        0        0 S..    .  -   Synchronized
0147 RW        0        0 RW 0066 NR        0        0 S..    .  -   Synchronized
0148 RW        0        0 RW 0067 NR        0        0 S..    .  -   Synchronized
0149 RW        0        0 RW 0068 NR        0        0 S..    .  -   Synchronized
014A RW        0        0 RW 0069 NR        0        0 S..    .  -   Synchronized
014B RW        0        0 RW 006A NR        0        0 S..    .  -   Synchronized
014C RW        0        0 RW 006B NR        0        0 S..    .  -   Synchronized
014D RW        0        0 RW 006C NR        0        0 S..    .  -   Synchronized
014E RW        0        0 RW 006D NR        0        0 S..    .  -   Synchronized
014F RW        0        0 RW 006E NR        0        0 S..    .  -   Synchronized

Total    ------- -------             ------- -------
Trks           0    1012                   0       0
MBs          0.0    31.6                 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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf establish` command initiates an incremental establish operation on SRDF pairs in the composite group that are not synchronized (that is, the suspended pairs on Symmetrix 3143).

```
symrdf -cg SRDF establish -noprompt

An RDF 'Incremental Establish' operation execution is in progress for
   composite group 'SRDF'. Please wait...

    Suspend RDF link(s) for device(s) in (3143,01)..................Done.
    Resume RDF link(s) for device(s) in (3143,01)..................Not Done.
    Merge track tables between source and target in (3143,01).......Started.
    Devices: 00F7-00F8 ........................................... Merged.
    Device: 00FA ................................................. Merged.
    Devices: 00FC-0101 ........................................... Merged.
    Devices: 0102-0107 ........................................... Merged.
    Devices: 0108-010D ........................................... Merged.
    Devices: 010E-010F ........................................... Merged.
    Merge track tables between source and target in (3143,01).......Done.
    Resume RDF link(s) for device(s) in (3143,01)...................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   composite group 'SRDF'.
```

- Another `symrdf query` command checks the state of the SRDF pairs and shows that the previously suspended pairs are now in the process of synchronizing (SyncInProg).

```
symrdf -cg SRDF query

Composite Group Name      :  SRDF
Composite Group Type      :  RDF1
Number of Symmetrix Units :   2
Number of RDF (RA) Groups :   2

Symmetrix ID                : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number       :   1 (A)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI       ST                     C   S
      A                   N        A                     o   u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv     n   s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA s   p   STATE
----------------------- -- ----------------------- ----- ------ ------------
00F7 RW       0       0 RW 0062 NR       0       0 S..   .  -   Synchronized
00FA RW       0      45 RW 0065 NR       0       0 S..   .  -   SyncInProg
00FC RW       0      46 RW 0067 NR       0       0 S..   .  -   SyncInProg
00FD RW       0      46 RW 0068 NR       0       0 S..   .  -   SyncInProg
00FE RW       0       1 RW 0069 NR       0       0 S..   .  -   SyncInProg
00FF RW       0       1 RW 006A NR       0       0 S..   .  -   SyncInProg
0100 RW       0       1 RW 006B NR       0       0 S..   .  -   SyncInProg
0101 RW       0       1 RW 006C NR       0       0 S..   .  -   SyncInProg
0102 RW       0      46 RW 006D NR       0       0 S..   .  -   SyncInProg
0103 RW       0      46 RW 006E NR       0       0 S..   .  -   SyncInProg
0104 RW       0       1 RW 006F NR       0       0 S..   .  -   SyncInProg
0105 RW       0      46 RW 0070 NR       0       0 S..   .  -   SyncInProg
0106 RW       0       1 RW 0071 NR       0       0 S..   .  -   SyncInProg
0107 RW       0       1 RW 0072 NR       0       0 S..   .  -   SyncInProg
0108 RW       0      46 RW 0073 NR       0       0 S..   .  -   SyncInProg
0109 RW       0       1 RW 0074 NR       0       0 S..   .  -   SyncInProg
010A RW       0       1 RW 0075 NR       0       0 S..   .  -   SyncInProg
010B RW       0      46 RW 0076 NR       0       0 S..   .  -   SyncInProg
010C RW       0       1 RW 0077 NR       0       0 S..   .  -   SyncInProg
010D RW       0       1 RW 0078 NR       0       0 S..   .  -   SyncInProg
010E RW       0       1 RW 0079 NR       0       0 S..   .  -   SyncInProg
010F RW       0       1 RW 007A NR       0       0 S..   .  -   SyncInProg
```

```
Symmetrix ID                : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID         : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                LI       ST                       C   S
      A                 N        A                       o   u
      T  R1 Inv  R2 Inv K        T  R1 Inv  R2 Inv       n   s    RDF Pair
Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDA   s   p    STATE
----------------------- -- ----------------------- ----- ------ ------------
0137 RW       0       0 RW 0056 NR       0       0 S..   .   -    Synchronized
013A RW       0       0 RW 0059 NR       0       0 S..   .   -    Synchronized
013C RW       0       0 RW 005B NR       0       0 S..   .   -    Synchronized
013D RW       0       0 RW 005C NR       0       0 S..   .   -    Synchronized
013E RW       0       0 RW 005D NR       0       0 S..   .   -    Synchronized
013F RW       0       0 RW 005E NR       0       0 S..   .   -    Synchronized
0140 RW       0       0 RW 005F NR       0       0 S..   .   -    Synchronized
0141 RW       0       0 RW 0060 NR       0       0 S..   .   -    Synchronized
0142 RW       0       0 RW 0061 NR       0       0 S..   .   -    Synchronized
0143 RW       0       0 RW 0062 NR       0       0 S..   .   -    Synchronized
0144 RW       0       0 RW 0063 NR       0       0 S..   .   -    Synchronized
0145 RW       0       0 RW 0064 NR       0       0 S..   .   -    Synchronized
0146 RW       0       0 RW 0065 NR       0       0 S..   .   -    Synchronized
0147 RW       0       0 RW 0066 NR       0       0 S..   .   -    Synchronized
0148 RW       0       0 RW 0067 NR       0       0 S..   .   -    Synchronized
0149 RW       0       0 RW 0068 NR       0       0 S..   .   -    Synchronized
014A RW       0       0 RW 0069 NR       0       0 S..   .   -    Synchronized
014B RW       0       0 RW 006A NR       0       0 S..   .   -    Synchronized
014C RW       0       0 RW 006B NR       0       0 S..   .   -    Synchronized
014D RW       0       0 RW 006C NR       0       0 S..   .   -    Synchronized
014E RW       0       0 RW 006D NR       0       0 S..   .   -    Synchronized
014F RW       0       0 RW 006E NR       0       0 S..   .   -    Synchronized

Total   ------- -------            ------- -------
Trks          0     380                  0       0
MBs         0.0    11.9                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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symcg show` command displays that the consistency state of the devices is currently Disabled.

`symcg show SRDF`

```
Composite Group Name:  SRDF

  Composite Group Type                               : RDF1
  Valid                                              : Yes
  CG in PowerPath                                    : Yes
  CG in GNS                                          : No

  Number of RDF (RA) Groups                          :    2
  Number of STD Devices                              :   44
  Number of BCV's (Locally-associated)               :    0
  Number of VDEV's (Locally-associated)              :    0
  Number of RBCV's (Remotely-associated STD-RDF)     :    0
```

```
Number of BRBCV's (Remotely-associated BCV-RDF)    :    0
Number of RRBCV's (Remotely-associated RBCV)       :    0

Number of Symmetrix Units (2):
   {

   1) Symmetrix ID                                     : 000000003143
      Microcode Version                                : 5267
      Number of STD Devices                            :   22
      Number of BCV's (Locally-associated)             :    0
      Number of VDEV's (Locally-associated)            :    0
      Number of RBCV's (Remotely-associated STD_RDF)   :    0
      Number of BRBCV's (Remotely-associated BCV-RDF):    0
      Number of RRBCV's (Remotely-associated RBCV)     :    0

      Number of RDF (RA) Groups (1):
         {

         1) RDF (RA) Group Number   :  1            (A)
            Remote Symmetrix ID     : 000000003156
            Microcode Version       : 5267

            STD Devices (22):
               {
              -------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
               PdevName                 Dev  Config         State        (MB)
              -------------------------------------------------------------
               /dev/vx/rdmp/c15t1d24s2 00F7 RDF1           Disabled     12946
               /dev/vx/rdmp/c15t1d25s2 00FA RDF1           Disabled      8631
               /dev/vx/rdmp/c15t1d26s2 00FC RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d27s2 00FD RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d28s2 00FE RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d29s2 00FF RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d30s2 0100 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d31s2 0101 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d32s2 0102 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d33s2 0103 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d34s2 0104 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d35s2 0105 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d36s2 0106 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d37s2 0107 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d38s2 0108 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d39s2 0109 RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d40s2 010A RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d41s2 010B RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d42s2 010C RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d43s2 010D RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d44s2 010E RDF1           Disabled      4315
               /dev/vx/rdmp/c15t1d45s2 010F RDF1           Disabled      4315
               }
         }

   2) Symmetrix ID                                     : 000187900035
      Microcode Version                                : 5670
      Number of STD Devices                            :   22
      Number of BCV's (Locally-associated)             :    0
      Number of VDEV's (Locally-associated)            :    0
      Number of RBCV's (Remotely-associated STD_RDF)   :    0
      Number of BRBCV's (Remotely-associated BCV-RDF):    0
      Number of RRBCV's (Remotely-associated RBCV)     :    0

      Number of RDF (RA) Groups (1):
         {

         1) RDF (RA) Group Number   :  1            (00)
```

```
                Remote Symmetrix ID     : 000187900041
                Microcode Version       : 5670

                STD Devices (22):
                   {
                  -------------------------------------------------------------
                                           Sym  Device        Consistency   Cap
                   PdevName                Dev  Config        State         (MB)
                  -------------------------------------------------------------
                   /dev/vx/rdmp/c15t2d24s2 0137 RDF1+Mir      Disabled     12946
                   /dev/vx/rdmp/c15t2d25s2 013A RDF1+Mir      Disabled      8631
                   /dev/vx/rdmp/c15t2d26s2 013C RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d27s2 013D RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d28s2 013E RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d29s2 013F RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d30s2 0140 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d31s2 0141 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d32s2 0142 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d33s2 0143 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d34s2 0144 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d35s2 0145 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d36s2 0146 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d37s2 0147 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d38s2 0148 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d39s2 0149 RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d40s2 014A RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d41s2 014B RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d42s2 014C RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d43s2 014D RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d44s2 014E RDF1+Mir      Disabled      4315
                   /dev/vx/rdmp/c15t2d45s2 014F RDF1+Mir      Disabled      4315
                   }
             }
      }
```

- The `symcg enable` command enables consistency protection for device pairs in the composite group.

`symcg -cg SRDF enable -noprompt`

```
A consistency 'Enable' operation execution is in progress for composite group
   'SRDF'. Please wait...

The consistency 'Enable' operation successfully executed for composite group
   'SRDF'.
```

- Another `symrdf query` command displays all pairs in the Synchronized state and an X in the “Cons” column to indicate that all pairs are now enabled for consistency protection.

symcg -cg SRDF query

```
Composite Group Name     :  SRDF
Composite Group Type     :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number       :  1 (A)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI       ST                      C  S
      A                  N        A                      o  u
      T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
00F7 RW       0       0 RW 0062 NR       0        0 S..   X  -   Synchronized
00FA RW       0       0 RW 0065 NR       0        0 S..   X  -   Synchronized
00FC RW       0       0 RW 0067 NR       0        0 S..   X  -   Synchronized
00FD RW       0       0 RW 0068 NR       0        0 S..   X  -   Synchronized
00FE RW       0       0 RW 0069 NR       0        0 S..   X  -   Synchronized
00FF RW       0       0 RW 006A NR       0        0 S..   X  -   Synchronized
0100 RW       0       0 RW 006B NR       0        0 S..   X  -   Synchronized
0101 RW       0       0 RW 006C NR       0        0 S..   X  -   Synchronized
0102 RW       0       0 RW 006D NR       0        0 S..   X  -   Synchronized
0103 RW       0       0 RW 006E NR       0        0 S..   X  -   Synchronized
0104 RW       0       0 RW 006F NR       0        0 S..   X  -   Synchronized
0105 RW       0       0 RW 0070 NR       0        0 S..   X  -   Synchronized
0106 RW       0       0 RW 0071 NR       0        0 S..   X  -   Synchronized
0107 RW       0       0 RW 0072 NR       0        0 S..   X  -   Synchronized
0108 RW       0       0 RW 0073 NR       0        0 S..   X  -   Synchronized
0109 RW       0       0 RW 0074 NR       0        0 S..   X  -   Synchronized
010A RW       0       0 RW 0075 NR       0        0 S..   X  -   Synchronized
010B RW       0       0 RW 0076 NR       0        0 S..   X  -   Synchronized
010C RW       0       0 RW 0077 NR       0        0 S..   X  -   Synchronized
010D RW       0       0 RW 0078 NR       0        0 S..   X  -   Synchronized
010E RW       0       0 RW 0079 NR       0        0 S..   X  -   Synchronized
010F RW       0       0 RW 007A NR       0        0 S..   X  -   Synchronized

Symmetrix ID                : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID         : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI       ST                      C  S
      A                  N        A                      o  u
      T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
0137 RW       0       0 RW 0056 NR       0        0 S..   X  -   Synchronized
013A RW       0       0 RW 0059 NR       0        0 S..   X  -   Synchronized
013C RW       0       0 RW 005B NR       0        0 S..   X  -   Synchronized
013D RW       0       0 RW 005C NR       0        0 S..   X  -   Synchronized
013E RW       0       0 RW 005D NR       0        0 S..   X  -   Synchronized
013F RW       0       0 RW 005E NR       0        0 S..   X  -   Synchronized
0140 RW       0       0 RW 005F NR       0        0 S..   X  -   Synchronized
0141 RW       0       0 RW 0060 NR       0        0 S..   X  -   Synchronized
0142 RW       0       0 RW 0061 NR       0        0 S..   X  -   Synchronized
0143 RW       0       0 RW 0062 NR       0        0 S..   X  -   Synchronized
0144 RW       0       0 RW 0063 NR       0        0 S..   X  -   Synchronized
0145 RW       0       0 RW 0064 NR       0        0 S..   X  -   Synchronized
```

```
0146 RW       0        0 RW 0065 NR       0        0 S..   X  -   Synchronized
0147 RW       0        0 RW 0066 NR       0        0 S..   X  -   Synchronized
0148 RW       0        0 RW 0067 NR       0        0 S..   X  -   Synchronized
0149 RW       0        0 RW 0068 NR       0        0 S..   X  -   Synchronized
014A RW       0        0 RW 0069 NR       0        0 S..   X  -   Synchronized
014B RW       0        0 RW 006A NR       0        0 S..   X  -   Synchronized
014C RW       0        0 RW 006B NR       0        0 S..   X  -   Synchronized
014D RW       0        0 RW 006C NR       0        0 S..   X  -   Synchronized
014E RW       0        0 RW 006D NR       0        0 S..   X  -   Synchronized
014F RW       0        0 RW 006E NR       0        0 S..   X  -   Synchronized

Total   ------- -------             ------- -------
Trks          0       0                   0       0
MBs         0.0     0.0                 0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

◆ Another `symcg show` command also displays that the consistency state of the devices is now Enabled.

```
symcg show SRDF

Composite Group Name:  SRDF

  Composite Group Type                                    : RDF1
  Valid                                                   : Yes
  CG in PowerPath                                         : Yes
  CG in GNS                                               : No

  Number of RDF (RA) Groups                               :    2
  Number of STD Devices                                   :   44
  Number of BCV's (Locally-associated)                    :    0
  Number of VDEV's (Locally-associated)                   :    0
  Number of RBCV's (Remotely-associated STD-RDF)          :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)         :    0
  Number of RRBCV's (Remotely-associated RBCV)            :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                        : 000000003143
        Microcode Version                                   : 5267
        Number of STD Devices                               :   22
        Number of BCV's (Locally-associated)                :    0
        Number of VDEV's (Locally-associated)               :    0
        Number of RBCV's (Remotely-associated STD_RDF)      :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)        :    0

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number  :  1             (A)
              Remote Symmetrix ID    : 000000003156
              Microcode Version      : 5267

              STD Devices (22):
                 {
                ------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                 PdevName               Dev  Config        State         (MB)
                ------------------------------------------------------------
                 /dev/vx/rdmp/c15t1d24s2 00F7 RDF1          Enabled      12946
                 /dev/vx/rdmp/c15t1d25s2 00FA RDF1          Enabled       8631
                 /dev/vx/rdmp/c15t1d26s2 00FC RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d27s2 00FD RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d28s2 00FE RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d29s2 00FF RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d30s2 0100 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d31s2 0101 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d32s2 0102 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d33s2 0103 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d34s2 0104 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d35s2 0105 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d36s2 0106 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d37s2 0107 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d38s2 0108 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d39s2 0109 RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d40s2 010A RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d41s2 010B RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d42s2 010C RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d43s2 010D RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d44s2 010E RDF1          Enabled       4315
                 /dev/vx/rdmp/c15t1d45s2 010F RDF1          Enabled       4315
```

```
                  }
            }

      2) Symmetrix ID                                        : 000187900035
         Microcode Version                                   : 5670
         Number of STD Devices                               :   22
         Number of BCV's (Locally-associated)                :    0
         Number of VDEV's (Locally-associated)               :    0
         Number of RBCV's (Remotely-associated STD_RDF)      :    0
         Number of BRBCV's (Remotely-associated BCV-RDF):    0
         Number of RRBCV's (Remotely-associated RBCV)        :    0

         Number of RDF (RA) Groups (1):
            {

            1) RDF (RA) Group Number   :  1           (00)
               Remote Symmetrix ID     : 000187900041
               Microcode Version       : 5670

               STD Devices (22):
                  {
                ------------------------------------------------------------
                                         Sym  Device        Consistency   Cap
                 PdevName                Dev  Config        State         (MB)
                ------------------------------------------------------------
                /dev/vx/rdmp/c15t2d24s2 0137 RDF1+Mir      Enabled      12946
                /dev/vx/rdmp/c15t2d25s2 013A RDF1+Mir      Enabled       8631
                /dev/vx/rdmp/c15t2d26s2 013C RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d27s2 013D RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d28s2 013E RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d29s2 013F RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d30s2 0140 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d31s2 0141 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d32s2 0142 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d33s2 0143 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d34s2 0144 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d35s2 0145 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d36s2 0146 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d37s2 0147 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d38s2 0148 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d39s2 0149 RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d40s2 014A RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d41s2 014B RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d42s2 014C RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d43s2 014D RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d44s2 014E RDF1+Mir      Enabled       4315
                /dev/vx/rdmp/c15t2d45s2 014F RDF1+Mir      Enabled       4315
                  }
            }
      }
```

- The `symrdf suspend` command attempts to suspend SRDF pairs in the composite group. The message about using the "force flag" is meant to ensure that you really want to stop the SRDF mirroring operation and end consistency protection.

`symrdf -cg SRDF suspend -noprompt`

```
An RDF 'Suspend' operation execution is in progress for composite group
   'SRDF'.
   Please wait...

Cannot proceed in the current RDF Consistency state except if the force flag
   is used.
```

- The `symrdf suspend` command with the the `-force` option successfully suspends the SRDF pairs in the composite group.

`symrdf -cg SRDF suspend -force -noprompt`

```
An RDF 'Suspend' operation execution is in progress for composite group
   'SRDF'.
   Please wait...

     Pend I/O on RDF link(s) for device(s) in (0035,01)..............Done.
     Pend I/O on RDF link(s) for device(s) in (3143,01)..............Done.
     Suspend RDF link(s) for devices in (0035).......................Done.
     Suspend RDF link(s) for devices in (3143).......................Done.

The RDF 'Suspend' operation successfully executed for composite group
   'SRDF'.
```

Example 7: Creating Concurrent Dynamic SRDF Pairs

This example is performed using Solutions Enabler version 6.0. It creates concurrent dynamic SRDF pairs while the Symmetrix array is running. In this example, the controlling host is connected to a local Symmetrix array (sid 000187400011). The local Symmetrix is connected via RDF links to two remote Symmetrix arrays (sid 000187400093 and sid 000000006201). The example uses two different RDF (RA) groups to achieve the connection between each local R1 device and its two remote R2 mirrors.

The example assumes that the Symmetrix-wide configuration parameters concurrent RDF, dynamic RDF, and concurrent dynamic RDF have been enabled in the local Symmetrix array. Dynamic RDF must also be enabled in the remote Symmetrix arrays.

- The `symcfg list` command displays current RDF (RA) groups that serve as RDF links to connect local Symmetrix 000187400011 to remote Symmetrix 000187400093 through director 4D.

`symcfg list -ra 4D -sid 11`

```
Symmetrix ID: 000187400011

                S Y M M E T R I X    R D F    D I R E C T O R S


                                                 Remote           Local    Remote
  Ident   Symb   Num  Slot   Type       Attr   SymmID           RA Grp   RA Grp   Status

  RF-4D   04D    52     4   RDF-R1        -    000187400093 27 (1A) 23 (16) Online
                                            -    000187400093 28 (1B) 24 (17)
                                            -    000187400093 29 (1C) 25 (18)
                                            -    000187400093 30 (1D) 26 (19)
                                            -    000187400093 31 (1E) 27 (1A)
```

- The following `symrdf addgrp` command creates a dynamic RDF group that represents another RDF link between Symmetrix 000187400011 and Symmetrix 000187400093. It adds dynamic RDF group 58 on the local Symmetrix, and RDF group 58 on the remote Symmetrix. You must specify a group label (grp58 in this case) that can be used when modifying or deleting the group. Creation of the dynamic RDF group includes director 4D from the local Symmetrix and 3C from the remote Symmetrix as the director end points of this connection.

  It is important to be aware of your network topology when creating dynamic RDF groups between two Symmetrix arrays. To create a dynamic RDF link (a connection) between RA directors, the director end points must be able to see each other through the Fibre Channel fabric. For example, a dynamic RDF link can be created between local and remote directors only if the Fibre Channel zoning is set up so that the two directors can see each other through the fabric.

```
symrdf -v addgrp -label grp58 -rdfg 58 -sid 187400011 -dir 4D
           -remote_rdfg 58 -remote_sid 000187400093 -remote_dir 3C -noprompt

An RDF Addgrp operation execution is in
progress for dynamic group 'grp58'. Please wait...

  Successfully Added Dynamic RDF Group 'grp58' for Symm: 000187400011
```

Note: For brevity, this command creates an RDF Group with only one connection (local director 4D to remote director 3C). Using recommended practice, you would have at least two sets of RA directors supporting the RDF group. For example, the previous command could include `-dir 4D,4C` and `-remote_dir 3C,3D`.

- Another `symcfg list` command verifies the logical connections from the local director (4D) point of view. Dynamic RDF group 58 has been added to both the local and remote Symmetrix arrays.

symcfg list -ra 4D -sid 11

```
Symmetrix ID: 000187400011

                S Y M M E T R I X    R D F    D I R E C T O R S


                                                Remote        Local    Remote
  Ident   Symb   Num  Slot  Type       Attr  SymmID        RA Grp   RA Grp   Status

  RF-4D   04D    52     4  RDF-R1       -    000187400093 27 (1A) 23 (16) Online
                                          -    000187400093 28 (1B) 24 (17)
                                          -    000187400093 29 (1C) 25 (18)
                                          -    000187400093 30 (1D) 26 (19)
                                          -    000187400093 31 (1E) 27 (1A)
                                          -    000187400093 58 (39) 58 (39)
```

- Another `symcfg list` displays current RDF (RA) groups that serve as RDF links to connect local Symmetrix 000187400011 to the second remote Symmetrix (000000006201) through director 13A.

symcfg list -ra 13A -sid 11

```
Symmetrix ID: 000187400011

                S Y M M E T R I X    R D F    D I R E C T O R S


                                                Remote        Local    Remote
  Ident   Symb   Num  Slot  Type       Attr  SymmID        RA Grp   RA Grp   Status

  RE-13A  13A    13    13  RDF-R1       -    000000006201 47 (2E) 47 (2E) Online
                                          -    000000006201 50 (31) 50 (31)
```

- The following `symrdf addgrp` command creates a dynamic RDF group that represents another RDF link between Symmetrix 000187400011 and Symmetrix 000000006201. It adds dynamic RDF group 51 on the local Symmetrix, and RDF group 51 on the remote Symmetrix. Creation of the local and remote RDF groups includes director 13A from the local Symmetrix and 13B from the remote Symmetrix.

  symrdf -v addgrp -label grp51 -rdfg 51 -sid 187400011 -dir 13A
  -remote_rdfg 51 -remote_sid 000000006201 -remote_dir 13B -noprompt

```
An RDF Addgrp operation execution is in
progress for dynamic group 'grp51'. Please wait...

  Successfully Added Dynamic RDF Group 'grp51' for Symm: 000187400011
```

Note: For brevity, this command creates an RDF Group with only one connection (local director 13A to remote director 13B). Using recommended practice, you would have at least two sets of RA directors supporting the RDF group. For example, the previous command could include `-dir 13A,13B` and `-remote_dir 13B,13C`.

- Another `symcfg list` command verifies the logical connections from the local director (13A) point of view. Dynamic RDF group 51 has been added to both the local and remote Symmetrix arrays.

```
symcfg list -ra 13A -sid 11

Symmetrix ID: 000187400011

                S Y M M E T R I X    R D F    D I R E C T O R S


                                             Remote        Local    Remote
  Ident   Symb   Num  Slot  Type       Attr  SymmID        RA Grp  RA Grp  Status

  RE-13A  13A    13    13  RDF-R1       -   000000006201 47 (2E) 47 (2E) Online
                                          -    000000006201 50 (31)  50 (31)
                                          -    000000006201 51 (32)  51 (32)
```

- The `symdev list` command with the `-dynamic` option displays those devices on the local Symmetrix that have been configured to be capable of dynamic RDF. These will be the source devices.

```
symdev list -dynamic -sid 11

Symmetrix ID: 000187400011

        Device Name           Directors                  Device
--------------------------- ------------- ---------------------------------
                                                                       Cap
Sym  Physical                SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- ---------------------------------

0300 /dev/vx/rdmp/c6t0d0s2  14C:0 16A:D8  RAID-5       N/Grp'd      RW     449
0301 /dev/vx/rdmp/c6t0d1s2  14C:0 02A:D8  RAID-5       N/Grp'd      RW     449
0302 /dev/vx/rdmp/c6t0d2s2  14C:0 01A:C4  RAID-5       N/Grp'd      RW     449
0303 /dev/vx/rdmp/c6t0d3s2  14C:0 15A:C0  RAID-5       N/Grp'd      RW     449
0304 /dev/vx/rdmp/c6t0d4s2  14C:0 16A:C3  RAID-5       N/Grp'd      RW     449
0305 /dev/vx/rdmp/c6t0d5s2  14C:0 02A:C3  RAID-5       N/Grp'd      RW     449
```

- The following command uses the vi text editor to create a text file named OEA2OEB2.list. As was done here, you can enter into the file those device names that will constitute one of the sets of dynamic SRDF pairs (those R1/R2 pairs for local Symetrix 000187400011 and remote Symmetrix 000187400093. The R1 devices are listed in first column, and the remote R2 devices are listed in the second column on the same line as their respective R1 source. Like the R1 devices, the R2 devices must also be non-RDF devices that have been set with the dynamic RDF attribute.

```
vi OEA2OEB2.list

300  080
301  081
302  082
303  083
304  084
305  085
```

- The following command uses the vi text editor to create a text file named OEA2O6A2.list. As was done here, you can enter into the file those device names that will constitute the second set of dynamic SRDF pairs (those R1/R2 pairs for local Symetrix 000187400011 and remote Symmetrix 000000006201).

  `vi OEA2O6A2.list`

```
300  150
301  151
302  152
303  153
304  154
305  155
```

- The `symrdf createpair` command parses the file called OEA2OEB2.list that defines the dynamic SRDF pairs and specifies that the column-1 devices in the file are R1 devices (–type RDF1) on the local Symmetrix (000187400011). Communication is via RDF group 58 (–rdfg 58), which was previously established as the RDF link to remote Symmetrix 000187400093.

`symrdf createpair -file OEA2OEB2.list -sid 11 -rdfg 58 -type rdf1 -establish -noprompt`

```
An RDF 'Create Pair' operation execution is in progress for device
file 'OEA2OEB2.list'. Please wait...

    Create RDF Pair in (0011,58)....................................Done.
    Mark target device(s) in (0011,58) for full copy from source....Started.
    Devices: 0300-0305 ........................................... Marked.
    Mark target device(s) in (0011,58) for full copy from source....Done.
    Merge track tables between source and target in (0011,58).......Started.
    Devices: 0300-0305 ........................................... Merged.
    Merge track tables between source and target in (0011,58).......Done.
    Resume RDF link(s) for device(s) in (0011,58)...................Started.
    Resume RDF link(s) for device(s) in (0011,58)...................Done.

The RDF 'Create Pair' operation successfully executed for device
file 'OEA2OEB2.list'.
```

- A second `symrdf createpair` command parses the file called OEA2O6A2.list that defines the dynamic SRDF pairs and specifies that the column-1 devices in the file are R1 devices (`-type RDF1`) on the local Symmetrix (000187400011). Communication is via RDF group 51 (`-rdfg 51`), which was previously established as the RDF link to remote Symmetrix 000000006201.

`symrdf createpair -file OEA2O6A2.list -sid 11 -rdfg 51 -type rdf1 -establish -noprompt`

```
An RDF 'Create Pair' operation execution is in progress for device
file 'OEA2O6A2.list'. Please wait...

    Create RDF Pair in (0011,51)....................................Done.
    Mark target device(s) in (0011,51) for full copy from source....Started.
    Devices: 0300-0305 ........................................... Marked.
    Mark target device(s) in (0011,51) for full copy from source....Done.
    Merge track tables between source and target in (0011,51).......Started.
    Devices: 0300-0305 ........................................... Merged.
    Merge track tables between source and target in (0011,51).......Done.
    Resume RDF link(s) for device(s) in (0011,51)...................Started.
    Resume RDF link(s) for device(s) in (0011,51)...................Done.

The RDF 'Create Pair' operation successfully executed for device
file 'OEA2O6A2.list'.
```

- The `symdg` command creates an RDF1 type device group named dynConc. The `symld` command adds to the group the R1 devices that were created on the local Symmetrix (sid 11).

  symdg create -type rdf1 dynConc
  symld -g dynConc -sid 11 -range 300:305 addall dev

- The `symrdf query -rdfg all` command displays the concurrent SRDF pairings for the local R1 devices in the device group dynConc. The `-rdfg` option allows you to see the SRDF pairs represented by both RDF (RA) groups. As shown, all concurrent pairs are in the Synchronized state.

symrdf -g dynConc query -rdfg all

```
Device Group (DG) Name             : dynConc
DG's Type                          : RDF1
DG's Symmetrix ID                  : 000187400011
Remote Symmetrix ID                : 000000006201
RDF (RA) Group Number              : 51 (32)
Remote Symmetrix ID                : 000187400093
RDF (RA) Group Number              : 58 (39)


       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device   Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001   0300 RW       0        0 RW 0080 WD       0        0 S..   Synchronized
             RW       0        0 RW 0150 WD       0        0 S..   Synchronized
DEV002   0301 RW       0        0 RW 0081 WD       0        0 S..   Synchronized
             RW       0        0 RW 0151 WD       0        0 S..   Synchronized
DEV003   0302 RW       0        0 RW 0082 WD       0        0 S..   Synchronized
             RW       0        0 RW 0152 WD       0        0 S..   Synchronized
DEV004   0303 RW       0        0 RW 0083 WD       0        0 S..   Synchronized
             RW       0        0 RW 0153 WD       0        0 S..   Synchronized
DEV005   0304 RW       0        0 RW 0084 WD       0        0 S..   Synchronized
             RW       0        0 RW 0154 WD       0        0 S..   Synchronized
DEV006   0305 RW       0        0 RW 0085 WD       0        0 S..   Synchronized
             RW       0        0 RW 0155 WD       0        0 S..   Synchronized

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

6

Implementing Consistency Protection Using RDF-ECA and RDF-MSC

This chapter provides SYMCLI examples for implementing consistency protection across one or more database management systems within an SRDF configuration using RDF Enginuity Consistency Assist (RDF-ECA) for synchronous mode and RDF Multi Session Consistency (RDF-MSC) for asynchronous mode.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

Example 1: Consistency Protection in ASYNC Mode

This example was performed using Solutions Enabler version 6.1. A host is connected to local Symmetrix 000190300150, which is RDF-connected to a remote Symmetrix array (000190300152). The RDF daemon is installed on the host. The example uses the SRDF/A devices from local RDF (RA) groups 25 and 26.

- The `symcfg list` command with the `-rdfg all` option displays a list of RDF (RA) groups on the source Symmetrix array connected to the local host. Beginning with Enginuity Version 5671, all RDF groups on a Symmetrix array are capable of SRDF/A operation. The RDFA "Flags C" column of RA groups 25 and 26 indicates N/A (-), which means these groups are not operating in async mode. The ellipsis (…) represents omitted output.

`symcfg list -rdfg all -sid 150`

```
Symmetrix ID : 000190300150

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

    Local              Remote                  Group                RDFA Info
-------------  --------------------  -----------------------  ---------------
         LL                                   Flags  Dir  Flags  Cycle
RA-Grp  (sec)  RA-Grp  SymmID        T    Name     LPDS  Cfg  CSRM   time  Pri
-------------  --------------------  -----------------------  -----  ----- -
 1 ( 0)    10   1 ( 0) 000190300152  D bp4        XX..  F-S  XAM-   30    33
 2 ( 1)    10   2 ( 1) 000190300152  D power1     XX..  F-S  -IS-   30    33
 3 ( 2)    10   4 ( 3) 000190300152  D power2     XX..  F-S  -IS-   30    33
 4 ( 3)    10   9 ( 8) 000190300152  D power3     XX..  F-S  -IS-   30    33
 8 ( 7)    10   8 ( 7) 000190300152  D dav3       XX..  F-S  -IS-   30    33
10 ( 9)    10  10 ( 9) 000190300180  D test       XX..  F-S  -IS-   30    33
11 ( A)    10   3 ( 2) 000190300152  D bp1        XX..  F-S  XAM-   30    33
21 (14)    10  21 (14) 000190300180  D snhe2121   XX.N  F-S  -IS-   30    33
22 (15)    10  22 (15) 000190300152  D snhi2222   XX.N  F-S  .IS-   30    33
23 (16)    10  23 (16) 000190300180  D snhe2323   XX.N  F-S  -IS-   30    33
24 (17)    10  24 (17) 000190300152  D snhi2424   XX.N  F-S  .IS-   30    33
25 (18)    10  25 (18) 000190300152  D grp25      XX..  F-S  -IS-   30    33
26 (19)    10  26 (19) 000190300152  D grp26      XX..  F-S  -IS-   30    33
30 (1D)    10  30 (1D) 000190300152  D dav1       XX..  F-S  -IS-   30    33
.................................................................................

Legend:
  ?                  : Unknown
  Group (T)ype       :  S = Static, D = Dynamic
  Director (C)onfig  :  F-S = Fibre-Switched, F-H = Fibre-Hub
                        G = GIGE, E = ESCON, T = T3, - = N/A
  Group Flags        :
      Prevent Auto (L)ink Recovery     : X = Enabled, . = Disabled
      Prevent RAs Online Upon (P)ower On: X = Enabled, . = Disabled
      Link (D)omino                    : X = Enabled, . = Disabled
      (S)TAR mode                      : N = Normal, R = Recovery, . = OFF
  RDFA Flags         :
      (C)onsistency : X = Enabled, . = Disabled, - = N/A
      (S)tatus      : A = Active, I = Inactive, - = N/A
      (R)DFA Mode   : S = Single-session, M = MSC, - = N/A
      (M)sc Cleanup : C = MSC Cleanup required, - = N/A
```

- The `symcg create` command creates an RDF1-type composite group named oracle. You must specify the `-rdf_consistency` option to make the group capable of being enabled for RDF consistency protection.

  `symcg create oracle -type rdf1 -rdf_consistency`

- The `symcg addall` command adds standard devices from the source Symmetrix array (`-sid` 150) to the composite group, using the `-rdfg` option to add all devices from RDF groups 25 and 26.

```
symcg -cg oracle -sid 150 -rdfg 25 addall dev
symcg -cg oracle -sid 150 -rdfg 26 addall dev
```

- The `symcg list` command displays a list of composite groups defined on this host. Only one composite group is defined, and it contains two devices, one in each RDF (RAG) group. Include the -rdf_consistency option to display only those groups that are in the RDF consistency database.

```
symcg list -rdf_consistency

                         C O M P O S I T E   G R O U P S

                                              Number of         Number of
    Name                   Type     Valid  Symms    RAGs     Devs    BCVs    VDEVs

    oracle                 RDF1     Yes        1       2        2       0        0
```

- The `symrdf set mode async` command sets the method of replication to Asynchronous for the devices in the composite group.

```
symrdf -cg oracle set mode async -noprompt

An RDF Set 'Asynchronous Mode' operation execution is in
progress for composite group 'oracle'. Please wait...

The RDF Set 'Asynchronous Mode' operation successfully executed
for composite group 'oracle'.
```

- The `symcg show` command displays configuration and status information about the composite group. The "A" and "S" entries in the "Flags S and R" columns of each device indicate that SRDF/A is now active but still operating in single-session mode (the Symmetrix controls SRDF/A session management). A period (.) in the "C" column means RDF consistency is not yet enabled. When consistency is enabled, then the entire composite group will be enabled for consistency protection. If the links are up when the enable is performed, then the RDF (RA) groups will go from single-session mode to MSC mode (the RDF daemon controls RDFA session management). If the links are not up when the enable is performed, then the RDF (RA) groups will go into MSC mode when the links are brought up through an operation such as establish or resume. "RDF consistency Protection Allowed" depends on your creating the composite group using the `-rdf_consistency` option.

```
symcg show oracle

Composite Group Name:  oracle

  Composite Group Type                                : RDF1
  Valid                                               : Yes
  CG in PowerPath                                     : No
  CG in GNS                                           : No
  RDF Consistency Protection Allowed                  : Yes
  RDF Consistency Mode                                : NONE
  Concurrent RDF                                      : No

  Number of RDF (RA) Groups                           :     2
  Number of STD Devices                               :     2
  Number of CRDF STD Devices                          :     0
  Number of BCV's (Locally-associated)                :     0
  Number of VDEV's (Locally-associated)               :     0
  Number of RVDEV's (Remotely-associated VDEV)        :     0
  Number of RBCV's (Remotely-associated STD-RDF)      :     0
```

```
Number of BRBCV's (Remotely-associated BCV-RDF)    :    0
Number of RRBCV's (Remotely-associated RBCV)       :    0

Number of Symmetrix Units (1):
  {

  1) Symmetrix ID                                    : 000190300150
     Microcode Version                               : 5771
     Number of STD Devices                           :    2
     Number of CRDF STD Devices                      :    0
     Number of BCV's (Locally-associated)            :    0
     Number of VDEV's (Locally-associated)           :    0
     Number of RVDEV's (Remotely-associated VDEV)    :    0
     Number of RBCV's (Remotely-associated STD_RDF)  :    0
     Number of BRBCV's (Remotely-associated BCV-RDF):    0
     Number of RRBCV's (Remotely-associated RBCV)    :    0


     Number of RDF (RA) Groups (2):
       {

       1) RDF (RA) Group Number  : 25            (18)
          Remote Symmetrix ID    : 000190300152
          Microcode Version      : 5771
          Recovery RA Group      : N/A            (N/A)
          RA Group Name          : N/A

          STD Devices (1):
            {

 ------------------------------------------------------------------------
                                     Sym  Device     Flags    Cap
 LdevName    PdevName                Dev  Config     Sts      CSR    (MB)

 ------------------------------------------------------------------------
  DEV001      N/A                     0232 RDF1+R-5   RW    .AS     14370
            }

       2) RDF (RA) Group Number  : 26            (19)
          Remote Symmetrix ID    : 000190300152
          Microcode Version      : 5771
          Recovery RA Group      : N/A            (N/A)
          RA Group Name          : N/A

          STD Devices (1):
            {

 ------------------------------------------------------------------------
                                     Sym  Device      Flags   Cap
  LdevName    PdevName               Dev  Config      Sts     CSR    (MB)

 ------------------------------------------------------------------------
  DEV002      N/A                     0242 RDF1+R-5   RW    .AS     14370
            }
       }
   }

Legend:
  RDFA Flags:
       C(onsistency)    : X = Enabled, . = Disabled, - = N/A
       (RDFA) S(tatus)  : A = Active, I = Inactive, - = N/A
       R(DFA Mode)      : S = Single-session mode, M = MSC mode, - = N/A
```

- The `symcg set -name` commands are optional with a composite group that contains all asynchronous devices (as group oracle does) or all synchronous devices. If the composite group contains both asynchronous and synchronous devices and you wish to enable for consistency protection, then you must use the `symcg set -name` option. This has more relevance with concurrent RDF when you want to control asynchronous RDF groups separately from synchronous RDF groups (refer to *Example 5: Consistency Protection for Concurrent RDF* on page 6-23).

  Setting a name such as oracleAsync for the two RDF groups (25 and 26) allows you to perform SRDF control operations on these RDF groups using this name instead of the composite group name. This example performs control operations on the composite group, not on the RDF group name.

  `symcg -cg oracle set -name oracleAsync -rdfg 150:25`
  `symcg -cg oracle set -name oracleAsync -rdfg 150:26`

- The `symrdf split` command splits the devices in the composite group named oracle.

```
symrdf -cg oracle split -noprompt

An RDF 'Split' operation execution is
in progress for composite group 'oracle'. Please wait...

    Suspend RDF link(s) for device(s) in (0150,26)..................Started.
    Suspend RDF link(s) for device(s) in (0150,25)..................Started.
    Suspend RDF link(s) for device(s) in (0150,25)..................Done.
    Suspend RDF link(s) for device(s) in (0150,26)..................Done.
    Read/Write Enable device(s) in (0150,25) on RA at target (R2)...Done.
    Read/Write Enable device(s) in (0150,26) on RA at target (R2)...Done.
    Suspend RDF link(s) for device(s) in (0150,26)..................Started.
    Suspend RDF link(s) for device(s) in (0150,26)..................Done.

The RDF 'Split' operation successfully executed for
composite group 'oracle'.
```

- The `symrdf query` command displays the state of the SRDF/A pairs for each RDF group in the composite group oracle. Each SRDF pair is in the Split state. Including the `-detail` option provides the most information.

```
symrdf -cg oracle query -detail

Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  NONE

RDFG Names:
    {
    RDFG Name                          : oracleAsync
    RDF Consistency Mode               : NONE
    }

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 26 (19) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                       : 0
    Session Status                     : Inactive
    Minimum Cycle Time                 : 00:00:30
    Avg Cycle Time                     : 00:00:00
    Duration of Last cycle             : 00:00:00
    Session Priority                   : 33
```

```
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1          : 00:00:00
    R1 Side Percent Cache In Use       :  0
    R2 Side Percent Cache In Use       :  0
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical   Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV002   0242 RW       0  294766 NR 0242 RW       0       0 A...   Split

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number              : 25 (18) - oracleAsync
Star Mode                          : NO
RDFA Info:
    {
    Cycle Number                       : 0
    Session Status                     : Inactive
    Minimum Cycle Time                 : 00:00:30
    Avg Cycle Time                     : 00:00:00
    Duration of Last cycle             : 00:00:00
    Session Priority                   : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1          : 00:00:00
    R1 Side Percent Cache In Use       :  0
    R2 Side Percent Cache In Use       :  0
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical   Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV001   0232 RW       0  294784 NR 0232 RW       0       0 A...   Split

Total             -------- -------             -------- -------
  Track(s)               0  589550                    0       0
  MBs                  0.0 18423.4                  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
```

- The following command initiates the synchronization of SRDF pairs in the composite group.

symrdf -cg oracle establish -noprompt

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

    Write Disable device(s) in (0150,26) on RA at target (R2).......Done.
    Write Disable device(s) in (0150,25) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (0150,26).................Started.
    Suspend RDF link(s) for device(s) in (0150,25).................Done.
    Suspend RDF link(s) for device(s) in (0150,26).................Done.
    Suspend RDF link(s) for device(s) in (0150,26).................Started.
    Suspend RDF link(s) for device(s) in (0150,26).................Done.
    Resume RDF link(s) for device(s) in (0150,26)..................Started.
    Resume RDF link(s) for device(s) in (0150,25)..................Started.
    Merge track tables between source and target in (0150,25).......Started.
    Devices: 0232-0241 ......................................... Merged.
    Merge track tables between source and target in (0150,26).......Started.
    Devices: 0242-0251 ......................................... Merged.
    Merge track tables between source and target in (0150,25).......Done.
    Merge track tables between source and target in (0150,26).......Done.
    Resume RDF link(s) for device(s) in (0150,25)..................Done.
    Resume RDF link(s) for device(s) in (0150,26)..................Done.

The RDF 'Incremental Establish' operation successfully initiated for
composite group 'oracle'.
```

- All pairs are in the process of becoming consistent (SyncInProg) and moving toward the Consistent state ("Consistent" is a state characteristic of SRDF/A pairs). The `symrdf verify` command checks the state of SRDF/A pairs in the composite group every 60 seconds until all pairs are in the Consistent state.

  symrdf -cg oracle -consistent verify -i 60

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

  None of the devices in the group 'oracle' are in 'Consistent' state.

  ..............................................................................
  Not all of the devices in the group 'oracle' are in 'Consistent' state.

  All devices in the group 'oracle' are in 'Consistent' state.
  ```

- The `symcg enable` command enables RDF consistency protection for the composite group. The group is now known as a consistency group. At this point, the RDF daemon takes over RDFA session management from the Symmetrix array.

  symcg -cg oracle enable -noprompt

  ```
  A consistency 'Enable' operation execution is
  in progress for composite group 'oracle'. Please wait...

  The consistency 'Enable' operation successfully executed for
  composite group 'oracle'.
  ```

- The `symrdf verify` command with the `-cg_consistent` option checks every 120 seconds to determine when the consistency group reaches the Consistent state. This state occurs when at least two cycle switches have occurred since all devices in each RDF (RA) group reached a consistent state.

  symrdf -cg oracle verify -cg_consistent -i 120

  ```
  CG 'oracle' is NOT RDF-Consistent.
  CG 'oracle' is RDF-Consistent.
  ```

- The `symrdf query` command with the `-detail` option shows that the consistency protection for the group is enabled and all SRDF/A device pairs are in the "Consistent" state. "RDFA MSC Consistency Info" shows that the "Consistency State" of the consistency group is also "CONSISTENT," meaning that some cycle switches have occurred since all devices in each RDF (RA) group reached the "Consistent" state. Note that the "RDFG Names" information is relevant only if you enable RDF consistency via the "RDFG Name" instead of via the "Composite Group Name."

`symrdf -cg oracle query -detail`

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  MSC
RDFA MSC Consistency Info:
    {
    Session Status                    : Active
    Consistency State                 : CONSISTENT
    }

RDFG Names:
    {
    RDFG Name                         : oracleAsync
    RDF Consistency Mode              : NONE
    }


Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 26 (19) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                      : 8
    Session Status                    : Active - MSC
    Minimum Cycle Time                : 00:00:30
    Avg Cycle Time                    : 00:00:31
    Duration of Last cycle            : 00:00:30
    Session Priority                  : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1         : 00:00:56
    R1 Side Percent Cache In Use      :  0
    R2 Side Percent Cache In Use      :  0
    }


        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical   Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV002    0242 RW       0       0 RW 0242 WD       0       0 A..X   Consistent

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 25 (18) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                      : 8
    Session Status                    : Active - MSC
    Minimum Cycle Time                : 00:00:30
```

```
    Avg Cycle Time                     : 00:00:31
    Duration of Last cycle             : 00:00:30
    Session Priority                   : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1          : 00:00:56
    R1 Side Percent Cache In Use       :  0
    R2 Side Percent Cache In Use       :  0
    }

        Source (R1) View            Target (R2) View       MODES
-------------------------------- ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical   Sym  T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device    Dev  E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV001   0232 RW       0       0 RW 0232 WD       0       0 A..X   Consistent

Total             -------- -------            -------- -------
  Track(s)              0       0                   0       0
  MBs                 0.0     0.0                 0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A
```

Example 2: Tripping a Consistency Group Automatically

This example is a continuation of Example 1. The link represented by RDF (RA) group 26 is disconnected to simulate an automatic trip (unplanned interruption) of the consistency group. I/O continues to occur on the local Symmetrix array.

- At this point, the link represented by RDF (RA) group 26 is "disconnected." The query checks the status of SRDF/A pairs. The Partitioned state indicates that a physical link is down between an R1 device and its R2 target device. If only one link goes down and the other stays up, the latter goes to the Suspended state, and consistency on that link is maintained. The RDF daemon recognizes the interruption and suspends the other RDF link in the consistency group (RDF group 25) in a manner that honors dependent write I/Os. Recall that consistency protection is suspended when one or more R1 devices in a consistency group cannot propagate data to their corresponding R2 target devices. Although consistency protection is temporarily suspended, "RDF consistency" remains enabled.

`symrdf -cg oracle query -detail`

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  MSC
RDFA MSC Consistency Info
    {
    Session Status                    : Inactive
    Consistency State                 : N/A
    }

RDFG Names
    {
    RDFG Name                         : oracleAsync
    RDF Consistency Mode              : NONE
    }


Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : N/A             (Microcode Version: N/A)
RDF (RA) Group Number             : 26 (19) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                      : 0
    Session Status                    : Inactive
    Minimum Cycle Time                : 00:00:30
    Avg Cycle Time                    : 00:00:00
    Duration of Last cycle            : 00:00:00
    Session Priority                  : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1         : 00:00:00
    R1 Side Percent Cache In Use      :  0
    R2 Side Percent Cache In Use      :  0
    }


        Source (R1) View             Target (R2) View       MODES
-------------------------------- ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
```

```
DEV002   0242 RW      0   35094 NR 0242 NA      NA      NA A..X   Partitioned

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number              : 25 (18) - oracleAsync
Star Mode                          : NO
RDFA Info:
    {
    Cycle Number                        : 119
    Session Status                      : Active - MSC
    Minimum Cycle Time                  : 00:00:30
    Avg Cycle Time                      : 00:00:30
    Duration of Last cycle              : 00:00:30
    Session Priority                    : 33
    Tracks not Committed to the R2 Side: 39484
    Time that R2 is behind R1           : 00:02:24
    R1 Side Percent Cache In Use        :  4
    R2 Side Percent Cache In Use        :  8
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV001   0232 RW       0        0 RW 0232 WD       0        0 A..X   Suspended

Total             -------- -------            -------- -------
  Track(s)               0   35094                   0       0
  MBs                  0.0  1096.7                 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
```

- At this point, the RA link is reconnected. Once the link is repaired, the RDF pair state changes from Partitioned to Suspended. The `symrdf query` command displays this revised state.

`symrdf -cg oracle query -detail`

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode       :  MSC
RDFA MSC Consistency Info
    {
    Session Status                      : Inactive
    Consistency State                   : N/A
    }

RDFG Names
    {
    RDFG Name                           : oracleAsync
    RDF Consistency Mode                : NONE
    }


Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
```

```
RDF (RA) Group Number                : 26 (19) - oracleAsync
Star Mode                            : NO
RDFA Info:
    {
    Cycle Number                          : 0
    Session Status                        : Inactive
    Minimum Cycle Time                    : 00:00:30
    Avg Cycle Time                        : 00:00:00
    Duration of Last cycle                : 00:00:00
    Session Priority                      : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1             : 00:00:00
    R1 Side Percent Cache In Use          :  0
    R2 Side Percent Cache In Use          :  0
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                   LI       ST
  Standard     A                    N        A
Logical  Sym   T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV002   0242 RW       0   35094 NR 0242 WD    4326       0 A..X   Suspended

Symmetrix ID                        : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID                 : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number                : 25 (18) - oracleAsync
Star Mode                            : NO

RDFA Info:
    {
    Cycle Number                          : 0
    Session Status                        : Inactive - MSC
    Minimum Cycle Time                    : 00:00:30
    Avg Cycle Time                        : 00:00:00
    Duration of Last cycle                : 00:00:00
    Session Priority                      : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1             : 00:00:00
    R1 Side Percent Cache In Use          :  1
    R2 Side Percent Cache In Use          :  4
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                   LI       ST
  Standard     A                    N        A
Logical  Sym   T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV001   0232 RW       0    8110 NR 0232 WD   10862       0 A..X   Suspended

Total             -------- -------            -------- -------
  Track(s)               0   43204              15188       0
  MBs                  0.0  1350.1              474.6     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
```

- The SRDF/A pairs remain in the Suspended state until you manually re-establish them. The establish action first initiates any cache cleanup that may have been needed because the physical links went down, resulting in the last cycle being committed or discarded from the cache. Then the SRDF/A session and consistency protection are automatically resumed.

```
symrdf -cg oracle establish -noprompt

An RDF 'Incremental Establish' operation execution is
in progress for composite group 'oracle'. Please wait...

    Suspend RDF link(s) for device(s) in (0150,26)..................Started.
    Suspend RDF link(s) for device(s) in (0150,25)..................Started.
    Suspend RDF link(s) for device(s) in (0150,26)..................Done.
    Mark target device(s) in (0150,25) to refresh from source.......Started.
    Devices: 0232-0239 ........................................... Marked.
    Devices: 023A-0241 ........................................... Marked.
    Mark target device(s) in (0150,26) to refresh from source.......Started.
    Devices: 024E-0250 ........................................... Marked.
    Mark target device(s) in (0150,26) to refresh from source.......Done.
    Mark target device(s) in (0150,25) to refresh from source.......Done.
    Merge track tables between source and target in (0150,25).......Started.
    Merge track tables between source and target in (0150,26).......Started.
    Devices: 024E-0250 ........................................... Merged.
    Merge track tables between source and target in (0150,26).......Done.
    Devices: 0232-0241 ........................................... Merged.
    Merge track tables between source and target in (0150,25).......Done.
    Suspend RDF link(s) for device(s) in (0150,26)..................Started.
    Suspend RDF link(s) for device(s) in (0150,26)..................Done.
    Resume RDF link(s) for device(s) in (0150,26)...................Started.
    Resume RDF link(s) for device(s) in (0150,25)...................Started.
    Resume RDF link(s) for device(s) in (0150,25)...................Done.
    Resume RDF link(s) for device(s) in (0150,26)...................Done.

The RDF 'Incremental Establish' operation successfully initiated for
composite group 'oracle'.
```

- The `symrdf verify` command with th `-cg_consistent` option checks every 120 seconds to determine when the consistency group reaches the Consistent state.

```
symrdf -cg oracle verify -cg_consistent -i 120

CG 'oracle' is NOT RDF-Consistent.

CG 'oracle' is NOT RDF-Consistent.

CG 'oracle' is NOT RDF-Consistent.

CG 'oracle' is NOT RDF-Consistent.

CG 'oracle' is RDF-Consistent.
```

Example 3: Tripping a Consistency Group Manually

This example continues from the end of Example 2 to determine if tripping the consistency group manually produces similar results to suspending consistency protection when an unplanned interruption occurs.

- The `symrdf suspend` command deactivates the consistency group. The -force parameter confirms that you really want to stop the SRDF mirroring operation and suspend consistency protection.

```
symrdf  -cg oracle suspend -force

An RDF 'Suspend' operation execution is
in progress for composite group 'oracle'. Please wait...

    Suspend RDF link(s) for device(s) in (0150,26)..................Started.
    Suspend RDF link(s) for device(s) in (0150,25)..................Started.
    Suspend RDF link(s) for device(s) in (0150,25)..................Done.
    Suspend RDF link(s) for device(s) in (0150,26)..................Done.

The RDF 'Suspend' operation successfully executed for
composite group 'oracle'.
```

- The following query with the `-detail` option shows that all R1 devices from the consistency group are in the Suspended state. Consistency protection is temporarily "Inactive" but remains enabled.

```
symrdf -cg oracle query -detail

Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  MSC
RDFA MSC Consistency Info
    {
    Session Status                    : Inactive
    Consistency State                 : N/A
    }

RDFG Names
    {
    RDFG Name                         : oracleAsync
    RDF Consistency Mode              : NONE
    }

Symmetrix ID                  : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID           : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number          : 26 (19) - oracleAsync
Star Mode                      : NO
RDFA Info:
    {
    Cycle Number                      : 0
    Session Status                    : Inactive
    Minimum Cycle Time                : 00:00:30
    Avg Cycle Time                    : 00:00:00
    Duration of Last cycle            : 00:00:00
    Session Priority                  : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1         : 00:00:00
    R1 Side Percent Cache In Use      :  0
    R2 Side Percent Cache In Use      :  0
    }
```

```
        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV002   0242 RW       0   74806 NR 0242 WD       0       0 A..X   Suspended

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number              : 25 (18) - oracleAsync
Star Mode                          : NO
RDFA Info:
    {
    Cycle Number                       : 0
    Session Status                     : Inactive
    Minimum Cycle Time                 : 00:00:30
    Avg Cycle Time                     : 00:00:00
    Duration of Last cycle             : 00:00:00
    Session Priority                   : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1          : 00:00:00
    R1 Side Percent Cache In Use       :  0
    R2 Side Percent Cache In Use       :  0
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV001   0232 RW       0   74824 NR 0232 WD       0       0 A..X   Suspended

Total                 -------- -------             -------- --------
  Track(s)                   0  149630                    0        0
  MBs                      0.0  4675.9                  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
```

- The `symrdf resume` command resumes the RDF links between the SRDF/A pairs in the consistency group and I/O traffic between the R1 devices and their paired R2 devices. Normal SRDF mirroring resumes. Consistency protection is automatically activated again upon resumption of the link.

`symrdf -cg oracle resume`

```
An RDF 'Resume' operation execution is
in progress for composite group 'oracle'. Please wait...

    Resume RDF link(s) for device(s) in (0150,26)...................Started.
    Resume RDF link(s) for device(s) in (0150,25)...................Started.
    Resume RDF link(s) for device(s) in (0150,25)...................Done.
    Resume RDF link(s) for device(s) in (0150,26)...................Done.

The RDF 'Resume' operation successfully executed for
composite group 'oracle'.
```

◆ The `symrdf query` command displays the state of the SRDF/A pairs, which are in the process of becoming consistent (the SyncInProg state to the Consistent state).

symrdf -cg oracle query -detail

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  MSC
RDFA MSC Consistency Info
    {
    Session Status                          : Active
    Consistency State                       : INCONSISTENT
    }

RDFG Names
    {
    RDFG Name                               : oracleAsync
    RDF Consistency Mode                    : NONE
    }

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 26 (19) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                            : 2
    Session Status                          : Active - MSC
    Minimum Cycle Time                      : 00:00:30
    Avg Cycle Time                          : 00:00:28
    Duration of Last cycle                  : 00:00:28
    Session Priority                        : 33
    Tracks not Committed to the R2 Side: 17274
    Time that R2 is behind R1               : 00:00:53
    R1 Side Percent Cache In Use            :  3
    R2 Side Percent Cache In Use            :  2
    }

        Source (R1) View             Target (R2) View       MODES
-------------------------------  ------------------------ ----- ----------
              ST                  LI       ST
  Standard     A                   N        A
Logical   Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv        RDF Pair
Device    Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDAC   STATE
------------------------------- -- ---------------------- ----- ----------
DEV002    0242 RW       0  143372 RW 0242 WD       0       0 A..X   SyncInProg

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 25 (18) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                            : 2
    Session Status                          : Active - MSC
    Minimum Cycle Time                      : 00:00:30
    Avg Cycle Time                          : 00:00:30
    Duration of Last cycle                  : 00:00:30
    Session Priority                        : 33
    Tracks not Committed to the R2 Side: 14882
    Time that R2 is behind R1               : 00:00:55
    R1 Side Percent Cache In Use            :  2
    R2 Side Percent Cache In Use            :  3
    }
```

```
        Source (R1) View           Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV001   0232 RW       0  147046 RW 0232 WD       0       0 A..X   SyncInProg

Total              ------- -------           ------- -------
  Track(s)               0  290418                 0       0
  MBs                  0.0  9075.6               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
```

- A subsequent `symrdf query` command shows that all SRDF device pairs are now in the "Consistent" state and that the consistency group is "CONSISTENT" again.

`symrdf -cg oracle query -detail`

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  MSC
RDFA MSC Consistency Info
    {
    Session Status                      : Active
    Consistency State                   : CONSISTENT
    }

RDFG Names
    {
    RDFG Name                           : oracleAsync
    RDF Consistency Mode                : NONE
    }


Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 26 (19) - oracleAsync
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                        : 38
    Session Status                      : Active - MSC
    Minimum Cycle Time                  : 00:00:30
    Avg Cycle Time                      : 00:00:30
    Duration of Last cycle              : 00:00:30
    Session Priority                    : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1           : 00:00:40
    R1 Side Percent Cache In Use        :  0
    R2 Side Percent Cache In Use        :  0
    }
```

```
        Source (R1) View            Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV002   0242 RW       0       0 RW 0242 WD       0       0 A..X   Consistent

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             : 25 (18) - oracleAsync
Star Mode                         : NO

RDFA Info:
    {
    Cycle Number                         : 38
    Session Status                       : Active - MSC
    Minimum Cycle Time                   : 00:00:30
    Avg Cycle Time                       : 00:00:30
    Duration of Last cycle               : 00:00:30
    Session Priority                     : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1            : 00:00:40
    R1 Side Percent Cache In Use         :  0
    R2 Side Percent Cache In Use         :  0
    }

        Source (R1) View            Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV001   0232 RW       0       0 RW 0232 WD       0       0 A..X   Consistent

Total            -------- -------           -------- -------
  Track(s)              0       0                  0       0
  MBs                 0.0     0.0                0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A
```

Example 4: Creating a Composite Group from Existing Sources

This example is performed using Solutions Enabler version 6.0. The example populates a composite group using devices from an existing device group.

- The `symdg list` command displays two device groups (ora1 and ora2) containing RDF devices that can be included in a composite group.

```
symdg list

                          D E V I C E     G R O U P S

                                                                 Number of
    Name                  Type      Valid  Symmetrix ID  Devs   GKs  BCVs  VDEVs

    ora1                  RDF1      Yes    000187400011    16     0     0      0
    ora2                  RDF1      Yes    000187400011    16     0     0      0
```

- The `symdg dg2cg` command creates and populates a composite group named oracle, using devices from a device group named ora1. The `-rdf_consistency` option creates the composite group in the host's RDF consistency database.

```
symdg dg2cg ora1 oracle -rdf_consistency

Adding STD device 0FB2 to group 'oracle'... OK
Adding STD device 0FB3 to group 'oracle'... OK
Adding STD device 0FB4 to group 'oracle'... OK
Adding STD device 0FB5 to group 'oracle'... OK
Adding STD device 0FB6 to group 'oracle'... OK
Adding STD device 0FB7 to group 'oracle'... OK
Adding STD device 0FB8 to group 'oracle'... OK
Adding STD device 0FB9 to group 'oracle'... OK
Adding STD device 0FBA to group 'oracle'... OK
Adding STD device 0FBB to group 'oracle'... OK
Adding STD device 0FBC to group 'oracle'... OK
Adding STD device 0FBD to group 'oracle'... OK
Adding STD device 0FBE to group 'oracle'... OK
Adding STD device 0FBF to group 'oracle'... OK
Adding STD device 0FC0 to group 'oracle'... OK
Adding STD device 0FC1 to group 'oracle'... OK

16 device(s) were added to group 'oracle'.
```

- The `symcg list` command displays a list of composite groups defined on this host. This display shows the new composite group and that 16 devices were added to it from the device group.

```
symcg list

                      C O M P O S I T E    G R O U P S

                                         Number of           Number of
    Name                    Type    Valid  Symms    RAGs    Devs   BCVs   VDEVs

    oracle                  RDF1    Yes        1       1      16      0       0
```

- This `symdg dg2cg` command adds devices from ora2 to the same composite group. You need to use the `-rename` option with this second dg2cg command because logical device names are carried from the device group to the composite group. In the case of default logical device names, those from ora2 can collide with the same logical device names from ora1. The `-rename` option generates new logical device names for the devices being added from a second device group.

`symdg dg2cg ora2 oracle -rdf_consistency -rename`

```
Adding STD device 0FC2 to group 'oracle'... OK
Adding STD device 0FC3 to group 'oracle'... OK
Adding STD device 0FC4 to group 'oracle'... OK
Adding STD device 0FC5 to group 'oracle'... OK
Adding STD device 0FC6 to group 'oracle'... OK
Adding STD device 0FC7 to group 'oracle'... OK
Adding STD device 0FC8 to group 'oracle'... OK
Adding STD device 0FC9 to group 'oracle'... OK
Adding STD device 0FCA to group 'oracle'... OK
Adding STD device 0FCB to group 'oracle'... OK
Adding STD device 0FCC to group 'oracle'... OK
Adding STD device 0FCD to group 'oracle'... OK
Adding STD device 0FCE to group 'oracle'... OK
Adding STD device 0FCF to group 'oracle'... OK
Adding STD device 0FD0 to group 'oracle'... OK
Adding STD device 0FD1 to group 'oracle'... OK

16 device(s) were added to group 'oracle'.
```

- This `symcg list` command shows that the oracle composite group now contains the 32 devices (16 from each device group).

`symcg list`

```
                      C O M P O S I T E   G R O U P S

                                                Number of           Number of
 Name                      Type     Valid  Symms     RAGs     Devs    BCVs     VDEVs

 oracle                    RDF1     Yes        1        2       32       0         0
```

- The `symcg show` command displays the details of the composite group oracle. Note that the logical device names of the ora2 devices (the second set of sixteen) were renamed in the composite group to be DEV017 through DEV032.

`symcg show oracle`

```
Composite Group Name:  oracle

  Composite Group Type                                  : RDF1
  Valid                                                 : Yes
  CG in PowerPath                                       : No
  CG in GNS                                             : No
  RDF Consistency Protection Allowed                    : Yes
  RDF Consistency Enabled                               : No

  Number of RDF (RA) Groups                             :    2
  Number of STD Devices                                 :   32
  Number of BCV's (Locally-associated)                  :    0
  Number of VDEV's (Locally-associated)                 :    0
  Number of RBCV's (Remotely-associated STD-RDF)        :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)       :    0
  Number of RRBCV's (Remotely-associated RBCV)          :    0

  Number of Symmetrix Units (1):
    {

    1)  Symmetrix ID                                     : 000187400011
        Microcode Version                                : 5671
        Number of STD Devices                            :   32
        Number of BCV's (Locally-associated)             :    0
        Number of VDEV's (Locally-associated)            :    0
        Number of RBCV's (Remotely-associated STD_RDF)   :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)     :    0

        Number of RDF (RA) Groups (2):
          {

          1) RDF (RA) Group Number  : 27            (1A)
             Remote Symmetrix ID    : 000187400093
             Microcode Version      : 5671


             STD Devices (16):
               {

   -------------------------------------------------------------------------
                                       Sym  Device     Flags   Cap
    LdevName    PdevName               Dev  Config     Sts     CSR       (MB)

   -------------------------------------------------------------------------
    DEV001      /dev/rdsk/c68t12d0     0FB2 RDF1         RW     .AS       449
    DEV002      /dev/rdsk/c68t12d1     0FB3 RDF1         RW     .AS       449
    DEV003      /dev/rdsk/c68t12d2     0FB4 RDF1         RW     .AS       449
    DEV004      /dev/rdsk/c68t12d3     0FB5 RDF1         RW     .AS       449
    DEV005      /dev/rdsk/c68t12d4     0FB6 RDF1         RW     .AS       449
    DEV006      /dev/rdsk/c68t12d5     0FB7 RDF1         RW     .AS       449
    DEV007      /dev/rdsk/c68t12d6     0FB8 RDF1         RW     .AS       449
    DEV008      /dev/rdsk/c68t12d7     0FB9 RDF1         RW     .AS       449
    DEV009      /dev/rdsk/c68t13d0     0FBA RDF1         RW     .AS       449
    DEV010      /dev/rdsk/c68t13d1     0FBB RDF1         RW     .AS       449
    DEV011      /dev/rdsk/c68t13d2     0FBC RDF1         RW     .AS       449
    DEV012      /dev/rdsk/c68t13d3     0FBD RDF1         RW     .AS       449
    DEV013      /dev/rdsk/c68t13d4     0FBE RDF1         RW     .AS       449
    DEV014      /dev/rdsk/c68t13d5     0FBF RDF1         RW     .AS       449
    DEV015      /dev/rdsk/c68t13d6     0FC0 RDF1         RW     .AS       449
```

```
 DEV016      /dev/rdsk/c68t13d7       0FC1 RDF1         RW     .AS      449
           }

       2) RDF (RA) Group Number  : 28             (1B)
          Remote Symmetrix ID    : 000187400093
          Microcode Version      : 5671

          STD Devices (16):
            {

--------------------------------------------------------------------------
                                      Sym Device       Flags   Cap
 LdevName    PdevName                 Dev Config       Sts     CSR    (MB)

--------------------------------------------------------------------------
 DEV017      /dev/rdsk/c68t14d0       0FC2 RDF1        RW     .AS      449
 DEV018      /dev/rdsk/c68t14d1       0FC3 RDF1        RW     .AS      449
 DEV019      /dev/rdsk/c68t14d2       0FC4 RDF1        RW     .AS      449
 DEV020      /dev/rdsk/c68t14d3       0FC5 RDF1        RW     .AS      449
 DEV021      /dev/rdsk/c68t14d4       0FC6 RDF1        RW     .AS      449
 DEV022      /dev/rdsk/c68t14d5       0FC7 RDF1        RW     .AS      449
 DEV023      /dev/rdsk/c68t14d6       0FC8 RDF1        RW     .AS      449
 DEV024      /dev/rdsk/c68t14d7       0FC9 RDF1        RW     .AS      449
 DEV025      /dev/rdsk/c68t15d0       0FCA RDF1        RW     .AS      449
 DEV026      /dev/rdsk/c68t15d1       0FCB RDF1        RW     .AS      449
 DEV027      /dev/rdsk/c68t15d2       0FCC RDF1        RW     .AS      449
 DEV028      /dev/rdsk/c68t15d3       0FCD RDF1        RW     .AS      449
 DEV029      /dev/rdsk/c68t15d4       0FCE RDF1        RW     .AS      449
 DEV030      /dev/rdsk/c68t15d5       0FCF RDF1        RW     .AS      449
 DEV031      /dev/rdsk/c68t15d6       0FD0 RDF1        RW     .AS      449
 DEV032      /dev/rdsk/c68t15d7       0FD1 RDF1        RW     .AS      449
           }
      }
  }

Legend:
  RDFA Flags:
       C(onsistency)    : X = Enabled, . = Disabled, - = N/A
       (RDFA) S(tatus)  : A = Active, I = Inactive, - = N/A
       R(DFA Mode)      : S = Single-session mode, M = MSC mode, - = N/A
```

Example 5: Consistency Protection for Concurrent RDF

This example is performed using Solutions Enabler version 6.1. The hardware configuration for the following concurrent RDF example consists of:

- Local Source Symmetrix (sid 150): R1 concurrent devices
- Remote Target Symmetrix (sid 180): R2 devices in synchronous mode
- Remote Target Symmetrix (sid 152): R2 devices to be run in asynchronous mode
- The `symrdf list` command with the `-concurrent` option shows devices on the local Symmetrix (sid 150) that are configured as concurrent RDF devices. Each of two remote devices of a concurrent R1 device belongs to a different RDF group (for example, "RDF Typ:G" 4 and 5). Device 0072 is the meta head of a 16-member meta device (0072 to 0081), and device 0082 is the meta head of a second meta device. These two meta head devices display the invalid tracks for all members of the meta device.

`symrdf list -sid 150 -concurrent`

```
Symmetrix ID: 000190300150

                              Local Device View
-------------------------------------------------------------------------
                   STATUS    MODES                       RDF  S T A T E S
Sym        RDF     --------- -----  R1 Inv   R2 Inv -----------------------
Dev  RDev  Typ:G   SA RA LNK  MDA   Tracks   Tracks Dev RDev Pair
---- ----  ------ --------- ----- -------  ------- --- ---- -------------

0072 0072   R1:4   RW RW RW   S..        0        0 RW  WD   Synchronized
     0072   R1:5   RW RW RW   S..        0        0 RW  WD   Synchronized
0073 0073   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0073   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0074 0074   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0074   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0075 0075   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0075   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0076 0076   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0076   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0077 0077   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0077   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0078 0078   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0078   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0079 0079   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0079   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007A 007A   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007A   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007B 007B   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007B   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007C 007C   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007C   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007D 007D   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007D   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007E 007E   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007E   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
007F 007F   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     007F   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0080 0080   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0080   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0081 0081   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0081   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
0082 0082   R1:4   RW RW RW   S..        0        0 RW  WD   Synchronized
     0082   R1:5   RW RW RW   S..        0        0 RW  WD   Synchronized
0083 0083   R1:4   RW RW RW   S..        -        - RW  WD   Synchronized
     0083   R1:5   RW RW RW   S..        -        - RW  WD   Synchronized
```

```
0084 0084   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0084   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0085 0085   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0085   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0086 0086   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0086   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0087 0087   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0087   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0088 0088   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0088   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0089 0089   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0089   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008A 008A   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008A   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008B 008B   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008B   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008C 008C   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008C   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008D 008D   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008D   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008E 008E   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008E   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
008F 008F   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     008F   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0090 0090   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0090   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized
0091 0091   R1:4   RW RW RW   S..          -         - RW   WD   Synchronized
     0091   R1:5   RW RW RW   S..          -         - RW   WD   Synchronized

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

- The following command creates a composite group named srdftest2. The `-type` parameter specifies an RDF1 type group (for the R1 devices). The `-rdf_consistency` parameter indicates that the composite group will be added to the RDF consistency database so that it can be managed for RDF consistency protection.

  `symcg create srdftest2 -type rdf1 -rdf_consistency`

- The following command adds to the composite group all devices from the two RDF groups that represent the concurrent links. With concurrent R1 devices, the command that adds one concurrent link (for example, RDF group 4) actually adds both concurrent links: RDF group 4 and RDF group 5.

  `symcg -cg srdftest2 -sid 150 addall dev -rdfg 4`

◆ The `symrdf query` command displays local and remote Symmetrix array information and the status of the SRDF pairs in the composite group. Currently, the SRDF pairs are in the "Synchronized" state. Both RDF groups are initially operating in Sync mode ("S"); RDF group 5 will be switched to Async mode later. You can use the `-detail` option to provide more information than the standard query operation. This is particularly useful if RDF group names have been set and when RDFA information is needed (as is shown later).

`symrdf -cg srdftest2 query -detail`

```
Composite Group Name      :  srdftest2
Composite Group Type      :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode      :  NONE

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300180    (Microcode Version: 5771)
RDF (RA) Group Number             :  4 (03)
Star Mode                         : NO

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI       ST
  Standard     A                   N        A
Logical   Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV001   0072 RW       0       0 RW 0072 WD       0       0 S...   Synchronized
DEV002   0082 RW       0       0 RW 0082 WD       0       0 S...   Synchronized

Symmetrix ID                     : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID              : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             :  5 (04)
Star Mode                         : NO

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI       ST
  Standard     A                   N        A
Logical   Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV001   0072 RW       0       0 RW 0072 WD       0       0 S...   Synchronized
DEV002   0082 RW       0       0 RW 0082 WD       0       0 S...   Synchronized

Total                 -------- -------             -------- -------
  Track(s)                  0       0                    0       0
  MBs                     0.0     0.0                  0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A
```

◆ The `symcg set -name` commands create names for each RDF group for use in controlling consistency via this name. For example, the names "rdfg4" for RDF group 4, and "rdfg5" for RDF group 5. Specify the `-rdfg` parameter and the "sid:rdfg" format ("150:4" means Symmetrix 150 and RDF group 4).

`symcg -cg srdftest2 set -name rdfg4 -rdfg 150:4`

`symcg -cg srdftest2 set -name rdfg5 -rdfg 150:5`

- The `symrdf set mode async` command sets the method of replication to asynchronous (ASYNC) for the devices in the composite group subset named rdfg5. This begins asynchronous replication to Symmetrix 152, while continuing synchronous replication to Symmetrix 180.

```
symrdf -cg srdftest2 -rdfg name:rdfg5 set mode async -noprompt

An RDF Set 'Asynchronous Mode' operation execution is in
progress for composite group 'srdftest2'. Please wait...

The RDF Set 'Asynchronous Mode' operation successfully executed
for composite group 'srdftest2'.
```

- Another `symrdf query` command shows that the SRDF pairs in rdfg5 (RDF group 5, Symmetrix 152) are now in Async mode ("A"), and their pair state is Consistent. Note that the `-detail` option provides "RDFG Names" information and "RDFA Info" details.

```
symrdf -cg srdftest2 query -detail

Composite Group Name       :  srdftest2
Composite Group Type       :  RDF1
Number of Symmetrix Units  :  1
Number of RDF (RA) Groups  :  2
RDF Consistency Mode       :  NONE

RDFG Names
    {
    RDFG Name                          : rdfg4
    RDF Consistency Mode               : NONE

    RDFG Name                          : rdfg5
    RDF Consistency Mode               : NONE
    }

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300180    (Microcode Version: 5771)
RDF (RA) Group Number             :  4 (03) - rdfg4
Star Mode                         : NO

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------ ----- ----------
              ST                   LI      ST
  Standard     A                    N       A
Logical   Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------ ----- ----------
DEV002    0082  RW       0       0 RW 0082  WD       0       0 S...  Synchronized
DEV001    0072  RW       0       0 RW 0072  WD       0       0 S...  Synchronized

Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number             :  5 (04) - rdfg5
Star Mode                         : NO
RDFA Info:
    {
    Cycle Number                       : 1
    Session Status                     : Active
    Minimum Cycle Time                 : 00:00:30
    Avg Cycle Time                     : 00:00:30
    Duration of Last cycle             : 00:00:00
    Session Priority                   : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1          : 00:00:21
    R1 Side Percent Cache In Use       :  0
    R2 Side Percent Cache In Use       :  0
    }
```

```
        Source (R1) View            Target (R2) View       MODES
-------------------------------- ------------------------- ----- ----------
              ST                   LI      ST
  Standard     A                    N       A
Logical  Sym   T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ------------------------- ----- ----------
DEV002   0082 RW       0       0 RW 0082 WD       0       0 A...   Consistent
DEV001   0072 RW       0       0 RW 0072 WD       0       0 A...   Consistent

Total            ------- -------            ------- -------
  Track(s)             0       0                  0       0
  MBs                0.0     0.0                0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A
```

- The `symcg enable` command enables RDF consistency protection for the subset rdfg4.

  symcg -cg srdftest2 -rdfg name:rdfg4 enable -noprompt

  ```
  A consistency 'Enable' operation execution is
  in progress for composite group 'srdftest2'. Please wait...

  The consistency 'Enable' operation successfully executed for
  composite group 'srdftest2'.
  ```

- Another `symcg enable` command enables RDF consistency protection for the subset rdfg5.

  symcg -cg srdftest2 -rdfg name:rdfg5 enable -noprompt

  ```
  A consistency 'Enable' operation execution is
  in progress for composite group 'srdftest2'. Please wait...

  The consistency 'Enable' operation successfully executed for
  composite group 'srdftest2'.
  ```

- Tripping a subset of the consistency group manually produces similar results to suspending consistency protection when an unplanned interruption occurs. The following `symrdf suspend` command deactivates the rdfg4 subset, thus suspending the synchronous link. The `-force` option is required here to ensure that you really want to stop the SRDF mirroring operation and suspend consistency protection on the synchronous link.

  symrdf -cg srdftest2 suspend -rdfg name:rdfg4 -noprompt -force

  ```
  An RDF 'Suspend' operation execution is
  in progress for composite group 'srdftest2'. Please wait...

      Pend I/O on RDF link(s) for device(s) in (0150,04)..............Done.
      Suspend RDF link(s) for device(s) in (0150,04)..................Done.

  The RDF 'Suspend' operation successfully executed for
  composite group 'srdftest2'.
  ```

- Another `symrdf query` command shows that the synchronous link is suspended but the asynchronous link is unaffected.

```
symrdf -cg srdftest2 query -detail

Composite Group Name       :  srdftest2
Composite Group Type       :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode       :  NONE

RDFG Names
    {
    RDFG Name                           : rdfg4
    RDF Consistency Mode                : SYNC
    Sync Consistency Info
        {
        Consistency State               : N/A
        }

    RDFG Name                           : rdfg5
    RDF Consistency Mode                : MSC
    MSC Consistency Info
        {
        Session Status                  : Active
        Consistency State               : Consistent
        }
    }

Symmetrix ID                       : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID                : 000190300180    (Microcode Version: 5771)
RDF (RA) Group Number               :  4 (03) - rdfg4
Star Mode                           : NO

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
Logical  Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device   Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
--------------------------------  -- ----------------------- ----- ----------
DEV002   0082 RW       0       0 NR 0082 WD       0       0 S..X   Suspended
DEV001   0072 RW       0       0 NR 0072 WD       0       0 S..X   Suspended

Symmetrix ID                       : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID                : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number               :  5 (04) - rdfg5
Star Mode                           : NO
RDFA Info:
    {
    Cycle Number                        : 39
    Session Status                      : Active - MSC
    Minimum Cycle Time                  : 00:00:30
    Avg Cycle Time                      : 00:00:30
    Duration of Last cycle              : 00:00:30
    Session Priority                    : 33
    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1           : 00:00:51
    R1 Side Percent Cache In Use        :  0
    R2 Side Percent Cache In Use        :  0
    }

        Source (R1) View             Target (R2) View       MODES
--------------------------------  ------------------------- ----- ----------
              ST                  LI      ST
  Standard     A                   N       A
```

```
Logical   Sym   T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       RDF Pair
Device    Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDAC  STATE
-------------------------------- -- ----------------------- ----- ----------
DEV002    0082 RW       0       0 RW 0082 WD       0       0 A..X  Consistent
DEV001    0072 RW       0       0 RW 0072 WD       0       0 A..X  Consistent

Total            -------- --------            -------- --------
  Track(s)              0       0                    0       0
  MBs                 0.0     0.0                  0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A
```

- The `symrdf resume` command resumes the synchronous RDF link represented by the rdfg4 subset of the consistency group. I/O traffic resumes between the synchronous R1 devices and their paired R2 devices. Normal SRDF mirroring resumes. Consistency protection is automatically activated upon resumption of the link.

`symrdf -cg srdftest2 resume -rdfg name:rdfg4 -noprompt`

```
An RDF 'Resume' operation execution is
in progress for composite group 'srdftest2'. Please wait...

    Resume RDF link(s) for device(s) in (0150,04)...................Started.
    Resume RDF link(s) for device(s) in (0150,04)...................Done.

The RDF 'Resume' operation successfully executed for
composite group 'srdftest2'.
```

- The `symrdf verify` command displays a message every 30 seconds until all SRDF pairs in the rdfg4 subset are synchronized.

`symrdf -cg srdftest2 verify -rdfg name:rdfg4 -i 30 -synchronized`

```
Not all devices in the RDF group 'srdftest2' are in the 'Synchronized' state.

Not all devices in the RDF group 'srdftest2' are in the 'Synchronized' state.

All devices in the group 'srdftest2' are in 'Synchronized' state.
```

- The `symrdf query` command confirms that the synchronous pairs in rdfg4 have returned to the Synchronized state.

symrdf -cg srdftest2 query

```
Composite Group Name       :  srdftest2
Composite Group Type       :  RDF1
Number of Symmetrix Units :  1
Number of RDF (RA) Groups :  2
RDF Consistency Mode       :  NONE


Symmetrix ID                      : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID               : 000190300180    (Microcode Version: 5771)
RDF (RA) Group Number              :  4 (03) - rdfg4
Star Mode                          : NO

Source (R1) View             Target (R2) View       MODES STATES
-------------------------------  ------------------------ ----- ------ --------
             ST                 LI      ST                        C  S
  Standard    A                  N       A                        o  u
Logical  Sym  T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv        n  s   RDF Pair
Device   Dev  E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA    s  p   STATE
------------------------------- -- ------------------------ ----- ------ --------
DEV002   0082 RW      0       0 RW 0082 WD       0       0 S..    X  -   Synchronized
DEV001   0072 RW      0       0 RW 0072 WD       0       0 S..    X  -   Synchronized


Symmetrix ID                    : 000190300150    (Microcode Version: 5771)
Remote Symmetrix ID             : 000190300152    (Microcode Version: 5771)
RDF (RA) Group Number           :  5 (04) - rdfg5
Star Mode                       : NO

       Source (R1) View             Target (R2) View       MODES STATES
-------------------------------  ------------------------ ----- ------ --------
             ST                 LI      ST                        C  S
  Standard    A                  N       A                        o  u
Logical  Sym  T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv        n  s   RDF Pair
Device   Dev  E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA    s  p   STATE
------------------------------- -- ------------------------ ----- ------ --------
DEV002   0082 RW      0       0 RW 0082 WD       0       0 A..    X  -   Consistent
DEV001   0072 RW      0       0 RW 0072 WD       0       0 A..    X  -   Consistent

Total             -------- --------             -------- --------
  Track(s)               0        0                    0        0
  MBs                  0.0      0.0                  0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

7

Implementing Consistency Protection Using PowerPath

This chapter provides examples for implementing consistency protection across one or more database management systems within an SRDF configuration using PowerPath.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

Example 1: Implementing Consistency Protection

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host connected to two Symmetrix arrays (Symmetrix 000000003143 and Symmetrix 000000003087). PowerPath 2.1.1 and Oracle 8.1.7.0.0 are installed on the host. The example uses PowerPath R1 devices 5D through 65 on both Symmetrix arrays. An Oracle database has been installed on the production host. All Oracle objects (data files, control files and redo logs) must be on the PowerPath devices.

- The `sympd list` command with the `-powerpath` option displays a list of host-visible PowerPath devices on the two Symmetrix arrays that are connected to this host. The display below shows a portion of this list.

`sympd list -powerpath`

```
Symmetrix ID: 000000003087

                      P O W E R P A T H    D E V I C E S

       Device Name           Directors                 Device
---------------------------- ------------ ---------------------------------
                                                                        Cap
Physical                   Sym  SA :P DA :IT Config        Attribute    Sts  (MB)
---------------------------- ------------ ---------------------------------

/dev/rdsk/emcpower52c    0000      01A:C0 2-Way Mir     N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d0s2       - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d0s2          - 14A:0      - -           -            -       -

/dev/rdsk/emcpower75c    005C      01B:D0 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d39s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d39s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower76c    005D      02B:D0 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d48s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d48s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower77c    005E      01A:C1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d49s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d49s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower78c    005F      02A:C1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d50s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d50s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower79c    0060      01B:C1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d51s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d51s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower80c    0061      02B:C1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d52s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d52s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower81c    0062      01A:D1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d53s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d53s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower82c    0063      02A:D1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d54s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d54s2         - 14A:0      - -           -            -       -

/dev/rdsk/emcpower83c    0064      01B:D1 RDF1+Mir      N/Grp'd      RW   4315
/dev/vx/rdmp/c2t0d55s2      - 14A:0      - -           -            -       -
/dev/rdsk/c2t0d55s2         - 14A:0      - -           -            -       -
```

```
/dev/rdsk/emcpower84c   0065       02B:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c2t0d64s2     - 14A:0      - -        -            -         -
/dev/rdsk/c2t0d64s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower102c  00EE       02A:C0 Unprotected   N/Grp'd      RW       3
/dev/vx/rdmp/c2t0d101s2    - 14A:0      - -        -            -         -
/dev/rdsk/c2t0d101s2       - 14A:0      - -        -            -         -

/dev/rdsk/emcpower103c  00EF       01B:D3 Unprotected   N/Grp'd      RW       3
/dev/vx/rdmp/c2t0d102s2    - 14A:0      - -        -            -         -
/dev/rdsk/c2t0d102s2       - 14A:0      - -        -            -         -
```

```
Symmetrix ID: 000000003143

                      P O W E R P A T H     D E V I C E S

       Device Name           Directors                Device
---------------------------- ------------ ----------------------------------
                                                                       Cap
Physical                   Sym  SA :P DA :IT Config       Attribute    Sts   (MB)
---------------------------- ------------ ----------------------------------

/dev/rdsk/emcpower23c   005C       02B:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d39s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d39s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower24c   005D       01B:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d48s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d48s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower25c   005E       02A:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d49s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d49s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower26c   005F       01A:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d50s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d50s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower27c   0060       02B:C1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d51s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d51s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower28c   0061       01B:C1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d52s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d52s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower29c   0062       02A:C1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d53s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d53s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower30c   0063       01A:C1 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d54s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d54s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower31c   0064       02B:D0 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d55s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d55s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower32c   0065       01B:D0 RDF1+Mir      N/Grp'd      RW    4315
/dev/vx/rdmp/c1t0d64s2     - 14A:0      - -        -            -         -
/dev/rdsk/c1t0d64s2        - 14A:0      - -        -            -         -

/dev/rdsk/emcpower49c   00BA       02B:C1 Unprotected   N/Grp'd      RW       3
/dev/vx/rdmp/c1t0d100s2    - 14A:0      - -        -            -         -
```

```
/dev/rdsk/c1t0d100s2       - 14A:0      - -           -          -       -

/dev/rdsk/emcpower50c   00BB      01A:D2 Unprotected   N/Grp'd      RW       3
/dev/vx/rdmp/c1t0d101s2    - 14A:0      - -           -          -       -
/dev/rdsk/c1t0d101s2       - 14A:0      - -           -          -       -
```

- The `symcg create` command creates a composite group named oracle on this host. Beginning with Solutions Enabler version 5.4, you must specify an RDF type to make the group capable of being enabled for consistency protection. Moreover, if you have not already set the `SYMAPI_RDF_CG_TO_PPATH` option to `ENABLE`, you must include the `-ppath` option so that the group is added to PowerPath.

  `symcg create oracle -type rdf1 -ppath`

The `symcg addall` commands add PowerPath standard devices from the two configured Symmetrix arrays to the composite group, using the `-range` option to limit the selections to those devices from 5D to 65.

```
symcg -cg oracle -sid 087 addall dev -range 5D:65
symcg -cg oracle -sid 143 addall dev -range 5D:65
```

- The `symcg list` command displays a list of composite groups defined on this host. Only one composite group is defined, and it contains eighteen devices, nine from each of the two configured Symmetrix arrays. Beginning with Solutions Enabler version 5.4, you can include the `-ppath` option to display only those groups that are in PowerPath.

`symcg list -ppath`

```
                     C O M P O S I T E   G R O U P S

                                            Number of          Number of
    Name                   Type    Valid  Symms    RAGs     Devs    BCVs    VDEVs

    oracle                 RDF1     Yes       2       2       18       0        0
```

- The `symcg show` command displays detailed configuration and status information about the composite group. Note that the current Consistency State of the devices is Disabled.

`symcg show oracle`

```
Composite Group Name:  oracle

  Composite Group Type                                   : RDF1
  Valid                                                   : Yes
  CG in PowerPath                                         : Yes
  CG in GNS                                               : No

  Number of RDF (RA) Groups                               :    2
  Number of STD Devices                                   :   18
  Number of BCV's (Locally-associated)                    :    0
  Number of VDEV's (Locally-associated)                   :    0
  Number of RBCV's (Remotely-associated STD-RDF)          :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)         :    0
  Number of RR BCV's (Remotely-associated RBCV)           :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                       : 000000003087
        Microcode Version                                  : 5670
        Number of STD Devices                              :    9
        Number of BCV's (Locally-associated)               :    0
        Number of VDEV's (Locally-associated)              :    0
        Number of RBCV's (Remotely-associated STD_RDF)     :    0
```

```
   Number of BRBCV's (Remotely-associated BCV-RDF):    0
   Number of RRBCV's (Remotely-associated RBCV)   :    0

   Number of RDF (RA) Groups (1):
      {

      1) RDF (RA) Group Number  :  1            (00)
         Remote Symmetrix ID    : 000000003003
         Microcode Version      : 5670

         Standard (STD) Devices (9):
            {
          -------------------------------------------------------------
                                   Sym  Device        Consistency   Cap
           PdevName                Dev  Config        State         (MB)
          -------------------------------------------------------------
          /dev/vx/rdmp/c2t0d48s2  005D RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d49s2  005E RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d50s2  005F RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d51s2  0060 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d52s2  0061 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d53s2  0062 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d54s2  0063 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d55s2  0064 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c2t0d64s2  0065 RDF1+Mir      Disabled      4315
            }
      }

2) Symmetrix ID                                   : 000000003143
   Microcode Version                              : 5670
   Number of STD Devices                          :    9
   Number of BCV's (Locally-associated)           :    0
   Number of VDEV's (Locally-associated)          :    0
   Number of RBCV's (Remotely-associated STD_RDF) :    0
   Number of BRBCV's (Remotely-associated BCV-RDF):    0
   Number of RRBCV's (Remotely-associated RBCV)   :    0

   Number of RDF (RA) Groups (1):
      {

      1) RDF (RA) Group Number  :  1            (00)
         Remote Symmetrix ID    : 000000003156
         Microcode Version      : 5670

         Standard (STD) Devices (9):
            {
          -------------------------------------------------------------
                                   Sym  Device        Consistency   Cap
           PdevName                Dev  Config        State         (MB)
          -------------------------------------------------------------
          /dev/vx/rdmp/c1t0d48s2  005D RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d49s2  005E RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d50s2  005F RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d51s2  0060 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d52s2  0061 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d53s2  0062 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d54s2  0063 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d55s2  0064 RDF1+Mir      Disabled      4315
          /dev/vx/rdmp/c1t0d64s2  0065 RDF1+Mir      Disabled      4315
            }
      }
}
```

- The `symcg enable` command enables consistency protection for device pairs in the composite group. The group is now known as a *consistency group*.

```
symcg -cg oracle enable -noprompt

A consistency 'Enable' operation execution is in progress for composite group
'oracle'. Please wait...

The composite group 'Enable' operation successfully executed for consistency
group 'oracle'.
```

- The `symcg show` command now displays that the device Consistency State is Enabled.

```
symcg show oracle

Composite Group Name:  oracle

  Composite Group Type                                  : RDF1
  Valid                                                  : Yes
  CG in PowerPath                                        : Yes
  CG in GNS                                              : No

  Number of RDF (RA) Groups                              :    2
  Number of STD Devices                                  :   18
  Number of BCV's (Locally-associated)                   :    0
  Number of VDEV's (Locally-associated)                  :    0
  Number of RBCV's (Remotely-associated STD-RDF)         :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)        :    0
  Number of RR BCV's (Remotely-associated RBCV)          :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                      : 000000003087
        Microcode Version                                 : 5670
        Number of STD Devices                             :    9
        Number of BCV's (Locally-associated)              :    0
        Number of VDEV's (Locally-associated)             :    0
        Number of RBCV's (Remotely-associated STD_RDF)    :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):     0
        Number of RRBCV's (Remotely-associated RBCV)      :    0

        Number of RDF (RA) Groups (00):
           {

           1) RDF (RA) Group Number  :  1             (00)
              Remote Symmetrix ID    : 000000003003
              Microcode Version      : 5670

              Standard (STD) Devices (9):
                 {
               ------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                PdevName                Dev  Config        State         (MB)
               ------------------------------------------------------------
               /dev/vx/rdmp/c2t0d48s2   005D RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d49s2   005E RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d50s2   005F RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d51s2   0060 RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d52s2   0061 RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d53s2   0062 RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d54s2   0063 RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d55s2   0064 RDF1+Mir      Enabled       4315
               /dev/vx/rdmp/c2t0d64s2   0065 RDF1+Mir      Enabled       4315
                 }
           }
```

```
2) Symmetrix ID                                     : 000000003143
   Microcode Version                                : 5670
   Number of STD Devices                            :    9
   Number of BCV's (Locally-associated)             :    0
   Number of VDEV's (Locally-associated)            :    0
   Number of RBCV's (Remotely-associated STD_RDF)   :    0
   Number of BRBCV's (Remotely-associated BCV-RDF):    0
   Number of RRBCV's (Remotely-associated RBCV)     :    0

   Number of RDF (RA) Groups (1):
      {

      1) RDF (RA) Group Number   :  1           (00)
         Remote Symmetrix ID     : 000000003156
         Microcode Version       : 5670

         Standard (STD) Devices (9):
            {
          -------------------------------------------------------------
                                   Sym  Device         Consistency   Cap
           PdevName                Dev  Config         State         (MB)
          -------------------------------------------------------------
          /dev/vx/rdmp/c1t0d48s2  005D RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d49s2  005E RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d50s2  005F RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d51s2  0060 RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d52s2  0061 RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d53s2  0062 RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d54s2  0063 RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d55s2  0064 RDF1+Mir       Enabled       4315
          /dev/vx/rdmp/c1t0d64s2  0065 RDF1+Mir       Enabled       4315
            }
      }
}
```

- The `symrdf query` command checks the state of the SRDF pairs. Note that all devices in the consistency group are in both the RDF pair Split state and the offline link suspend state (indicated by a "." in the "Susp" column. Both the R1 and R2 devices can be accessed for read/write activity (RW) by their respective hosts. An "X" in the "Cons" column indicates that the devices are enabled for consistency protection.

`symrdf -cg oracle query`

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID         : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI       ST                     C  S
      A                   N        A                     o  u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv     n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW        0     354 NR 0045 RW        0        0 S..  X  .  Split
005E RW        0     360 NR 0046 RW        0        0 S..  X  .  Split
005F RW        0     809 NR 0047 RW        0        0 S..  X  .  Split
0060 RW        0     798 NR 0048 RW        0        0 S..  X  .  Split
0061 RW        0       0 NR 0049 RW        0        0 S..  X  .  Split
0062 RW        0       0 NR 004A RW        0        0 S..  X  .  Split
0063 RW        0       0 NR 004B RW        0        0 S..  X  .  Split
```

```
0064 RW       0        0 NR 004C RW       0        0 S..   X  .   Split
0065 RW       0        0 NR 004D RW       0        0 S..   X  .   Split

Symmetrix ID                 : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID          : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number        :  1 (00)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                LI       ST                       C  S
      A                 N        A                       o  u
      T  R1 Inv  R2 Inv K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0      227 NR 0045 RW       0        0 S..   X  .   Split
005E RW       0      359 NR 0046 RW       0        0 S..   X  .   Split
005F RW       0      371 NR 0047 RW       0        0 S..   X  .   Split
0060 RW       0      127 NR 0048 RW       0        0 S..   X  .   Split
0061 RW       0      114 NR 0049 RW       0        0 S..   X  .   Split
0062 RW       0       76 NR 004A RW       0        0 S..   X  .   Split
0063 RW       0       90 NR 004B RW       0        0 S..   X  .   Split
0064 RW       0        0 NR 004C RW       0        0 S..   X  .   Split
0065 RW       0        0 NR 004D RW       0        0 S..   X  .   Split

Total   ------- -------            ------- -------
Trks          0    3685                  0       0
MBs         0.0   115.0                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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf establish` command initiates the synchronization of SRDF pairs in the consistency group. In the process, Enginuity compares the track tables of each source Symmetrix array and its target. If the track tables are not identical, the tables are merged. There are many reasons why track tables might change while the SRDF pairs are split, including new I/O to either the R1 side or R2 side.

symrdf -cg oracle establish -noprompt

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

    Write Disable device(s) in (3087,01) on RA at target (R2).......Done.
    Write Disable device(s) in (3143,01) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (3087,01).................Done.
    Suspend RDF link(s) for device(s) in (3143,01).................Done.
    Resume RDF link(s) for device(s) in (3087,01)..................Done.
    Resume RDF link(s) for device(s) in (3143,01)..................Not Done.
    Merge track tables between source and target in (3143,01).......Started.
    Device: 005D ................................................. Merged.
    Device: 005E ................................................. Merged.
    Device: 005F ................................................. Merged.
    Device: 0060 ................................................. Merged.
    Device: 0061 ................................................. Merged.
    Device: 0062 ................................................. Merged.
    Device: 0063 ................................................. Merged.
    Device: 0064 ................................................. Merged.
```

```
    Device: 0065 ............................................... Merged.
    Merge track tables between source and target in (3143,01).......Done.
    Resume RDF link(s) for device(s) in (3143,01)..................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   composite
   group 'oracle'.
```

- The `symrdf query` command displays the state of the SRDF pairs. Some pairs are in the process of synchronizing (SyncInProg), while others have completed synchronizing (Synchronized).

`symrdf -cg oracle query`

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID         : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ------------------------ ----- ------ ------------
005D RW       0     166 RW 0045 WD       0        0 S..   X  -   SyncInProg
005E RW       0     169 RW 0046 WD       0        0 S..   X  -   SyncInProg
005F RW       0     344 RW 0047 WD       0        0 S..   X  -   SyncInProg
0060 RW       0     338 RW 0048 WD       0        0 S..   X  -   SyncInProg
0061 RW       0       0 RW 0049 WD       0        0 S..   X  -   Synchronized
0062 RW       0       0 RW 004A WD       0        0 S..   X  -   Synchronized
0063 RW       0       0 RW 004B WD       0        0 S..   X  -   Synchronized
0064 RW       0       0 RW 004C WD       0        0 S..   X  -   Synchronized
0065 RW       0       0 RW 004D WD       0        0 S..   X  -   Synchronized

Symmetrix ID                : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ------------------------ ----- ------ ------------
005D RW       0     127 RW 0045 WD       0        0 S..   X  -   SyncInProg
005E RW       0     165 RW 0046 WD       0        0 S..   X  -   SyncInProg
005F RW       0     154 RW 0047 WD       0        0 S..   X  -   SyncInProg
0060 RW       0     511 RW 0048 WD       0        0 S..   X  -   SyncInProg
0061 RW       0     506 RW 0049 WD       0        0 S..   X  -   SyncInProg
0062 RW       0      37 RW 004A WD       0        0 S..   X  -   SyncInProg
0063 RW       0      44 RW 004B WD       0        0 S..   X  -   SyncInProg
0064 RW       0       0 RW 004C WD       0        0 S..   X  -   Synchronized
0065 RW       0       0 RW 004D WD       0        0 S..   X  -   Synchronized

Total    ------- -------             ------- -------
Trks           0    2561                   0       0
MBs          0.0    80.0                 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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf verify` command checks the state of the SRDF pairs in the consistency group every five seconds until the pairs are synchronized. Then the verify loop ends.

  `symrdf -cg oracle -synchronized verify -i 5`

  ```
  NOT all of the mirrored pairs are in the 'Synchronized' state.

  NOT all of the mirrored pairs are in the 'Synchronized' state.

  NOT all of the mirrored pairs are in the 'Synchronized' state.

  NOT all of the mirrored pairs are in the 'Synchronized' state.

  All devices in the CG group 'oracle' are in the 'Synchronized' state.
  ```

- The `symrdf split` command trips the consistency group, creating a DBMS-restartable copy of the database on the R2 target devices. After the split completes, the R2 devices are enabled for both reads and writes by target-side hosts. The `-force` option is required here to ensure that you really want to stop the SRDF mirroring and end consistency protection.

`symrdf -cg oracle split -noprompt -force`

```
An RDF 'Split' operation execution is in progress for composite group
   'oracle'.
   Please wait...

     Pend I/O on RDF link(s) for device(s) in (3087,01).............Done.
     Pend I/O on RDF link(s) for device(s) in (3143,01).............Done.
     Read/Write Enable device(s) in (3087,01) on RA at target (R2)...Done.
     Read/Write Enable device(s) in (3143,01) on RA at target (R2)...Done.

The RDF 'Split' operation successfully executed for composite group 'oracle'.
```

Example 2: Tripping a Consistency Group Automatically

This example is a continuation of Example 1. One of the two Symmetrix arrays is disconnected to cause an automatic trip of the consistency group. I/O is occurring on both Symmetrix arrays.

- The `symrdf query` command shows that all devices in the composite group are enabled for consistency protection (indicated by an “X” in the “Cons” column) and are in both the RDF pair Split state and the offline link suspend state (indicated by a “.” in the “Susp” column).

`symrdf -cg oracle query`

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     582 NR 0045 RW       0       0 S..   X  .   Split
005E RW       0     587 NR 0046 RW       0       0 S..   X  .   Split
005F RW       0     304 NR 0047 RW       0       0 S..   X  .   Split
0060 RW       0     302 NR 0048 RW       0       0 S..   X  .   Split
0061 RW       0       0 NR 0049 RW       0       0 S..   X  .   Split
0062 RW       0       0 NR 004A RW       0       0 S..   X  .   Split
0063 RW       0       0 NR 004B RW       0       0 S..   X  .   Split
0064 RW       0       0 NR 004C RW       0       0 S..   X  .   Split
0065 RW       0       0 NR 004D RW       0       0 S..   X  .   Split

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0    1049 NR 0045 RW       0       0 S..   X  .   Split
005E RW       0      42 NR 0046 RW       0       0 S..   X  .   Split
005F RW       0      45 NR 0047 RW       0       0 S..   X  .   Split
0060 RW       0    3219 NR 0048 RW       0       0 S..   X  .   Split
0061 RW       0    3208 NR 0049 RW       0       0 S..   X  .   Split
0062 RW       0     202 NR 004A RW       0       0 S..   X  .   Split
0063 RW       0     187 NR 004B RW       0       0 S..   X  .   Split
0064 RW       0       0 NR 004C RW       0       0 S..   X  .   Split
0065 RW       0       0 NR 004D RW       0       0 S..   X  .   Split

Total    ------- -------              ------- -------
Trks           0    9727                    0       0
MBs          0.0   303.0                  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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf establish` command initiates the synchronization of SRDF pairs in the consistency group.

symrdf -cg oracle establish -noprompt

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

    Write Disable device(s) in (3087,01) on RA at target (R2).......Done.
    Write Disable device(s) in (3143,01) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (3087,01)..................Done.
    Suspend RDF link(s) for device(s) in (3143,01)..................Done.
    Resume RDF link(s) for device(s) in (3087,01)...................Done.
    Resume RDF link(s) for device(s) in (3143,01)...................Not Done.
    Merge track tables between source and target in (3143,01).......Started.
    Device: 005D .................................................. Merged.
    Device: 005E .................................................. Merged.
    Device: 005F .................................................. Merged.
    Device: 0060 .................................................. Merged.
    Device: 0061 .................................................. Merged.
    Device: 0062 .................................................. Merged.
    Device: 0063 .................................................. Merged.
    Device: 0064 .................................................. Merged.
    Device: 0065 .................................................. Merged.
    Merge track tables between source and target in (3143,01).......Done.
    Resume RDF link(s) for device(s) in (3143,01)...................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   composite
   group 'oracle'.
```

- The `symrdf verify` command checks the state of the SRDF pairs in the consistency group every five seconds until the pairs are synchronized.

  symrdf -cg oracle -synchronized verify -i 5

```
NOT all of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

All devices in the CG group 'oracle' are in the 'Synchronized' state.
```

- At this point, the RA connections on one of the Symmetrix arrays (sid 087) are "unplugged." The `query` checks the status of SRDF pairs. The Partitioned state indicates that a physical link is down between an R1 device and its R2 target device. Recall that an automatic trip occurs when one or more R1 source devices in a consistency group cannot propagate data to their corresponding R2 target devices. Because R1 devices on one Symmetrix cannot propagate data because of lost physical connections, PowerPath automatically suspends I/O propagation to R2 devices in the second Symmetrix array.

symrdf -cg oracle query

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI       ST                       C  S
      A                  N        A                       o  u
      T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     137 NR 0045 NA       NA       NA S..   X  .   Partitioned
005E RW       0     142 NR 0046 NA       NA       NA S..   X  .   Partitioned
005F RW       0     519 NR 0047 NA       NA       NA S..   X  .   Partitioned
0060 RW       0     532 NR 0048 NA       NA       NA S..   X  .   Partitioned
0061 RW       0       0 NR 0049 NA       NA       NA S..   X  .   Partitioned
0062 RW       0       0 NR 004A NA       NA       NA S..   X  .   Partitioned
0063 RW       0       0 NR 004B NA       NA       NA S..   X  .   Partitioned
0064 RW       0       0 NR 004C NA       NA       NA S..   X  .   Partitioned
0065 RW       0       0 NR 004D NA       NA       NA S..   X  .   Partitioned

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI       ST                       C  S
      A                  N        A                       o  u
      T  R1 Inv  R2 Inv  K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     171 NR 0045 WD        0        0 S..   X  .   Suspended
005E RW       0     275 NR 0046 WD        0        0 S..   X  .   Suspended
005F RW       0     275 NR 0047 WD        0        0 S..   X  .   Suspended
0060 RW       0     120 NR 0048 WD        0        0 S..   X  .   Suspended
0061 RW       0     113 NR 0049 WD        0        0 S..   X  .   Suspended
0062 RW       0      57 NR 004A WD        0        0 S..   X  .   Suspended
0063 RW       0      74 NR 004B WD        0        0 S..   X  .   Suspended
0064 RW       0       0 NR 004C WD        0        0 S..   X  .   Suspended
0065 RW       0       0 NR 004D WD        0        0 S..   X  .   Suspended

Total    ------- -------              ------- -------
Trks           0    2415                    0       0
MBs          0.0    75.0                  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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)     : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- At this point, the RA connections are re-established (plugged back in). Once the link is repaired, the state of the SRDF pairs on the reconnected Symmetrix array change from Partitioned to Suspended. The `symrdf query` command displays this revised state.

symrdf -cg oracle query

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
----------------------- ------------------------- ----- ------ ------------
     ST                  LI      ST                      C  S
      A                   N       A                      o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0      434 NR 0045 WD       0       0 S..  X  .   Suspended
005E RW       0      452 NR 0046 WD       0       0 S..  X  .   Suspended
005F RW       0     1341 NR 0047 WD       0       0 S..  X  .   Suspended
0060 RW       0     1334 NR 0048 WD       0       0 S..  X  .   Suspended
0061 RW       0        0 NR 0049 WD       0       0 S..  X  .   Suspended
0062 RW       0        0 NR 004A WD       0       0 S..  X  .   Suspended
0063 RW       0        0 NR 004B WD       0       0 S..  X  .   Suspended
0064 RW       0        0 NR 004C WD       0       0 S..  X  .   Suspended
0065 RW       0        0 NR 004D WD       0       0 S..  X  .   Suspended

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
----------------------- ------------------------- ----- ------ ------------
     ST                  LI      ST                      C  S
      A                   N       A                      o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0      260 NR 0045 WD       0       0 S..  X  .   Suspended
005E RW       0      440 NR 0046 WD       0       0 S..  X  .   Suspended
005F RW       0      434 NR 0047 WD       0       0 S..  X  .   Suspended
0060 RW       0      120 NR 0048 WD       0       0 S..  X  .   Suspended
0061 RW       0      113 NR 0049 WD       0       0 S..  X  .   Suspended
0062 RW       0       92 NR 004A WD       0       0 S..  X  .   Suspended
0063 RW       0      115 NR 004B WD       0       0 S..  X  .   Suspended
0064 RW       0        0 NR 004C WD       0       0 S..  X  .   Suspended
0065 RW       0        0 NR 004D WD       0       0 S..  X  .   Suspended
Total    -------- --------            -------- -------
Trks          0     5135                    0       0
MBs         0.0    160.0                  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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)     : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The SRDF pairs remain in the Suspended state until you manually re-establish them.

  symrdf -cg oracle establish -noprompt

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

     Suspend RDF link(s) for device(s) in (3087,01)..................Done.
     Suspend RDF link(s) for device(s) in (3143,01)..................Done.
     Resume RDF link(s) for device(s) in (3087,01)...................Done.
     Resume RDF link(s) for device(s) in (3143,01)...................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   consistency
   group 'oracle'.
```

- The `symrdf query` command displays the state of the SRDF pairs. Some pairs are in the process of synchronizing (SyncInProg), while others have completed synchronizing (Synchronized).

symrdf -cg oracle query

```
Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                LI       ST                       C  S
      A                 N        A                       o  u
      T  R1 Inv  R2 Inv K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0      920 RW 0045 WD       0        0 S..   X  -   SyncInProg
005E RW       0     1284 RW 0046 WD       0        0 S..   X  -   SyncInProg
005F RW       0     6061 RW 0047 WD       0        0 S..   X  -   SyncInProg
0060 RW       0     6029 RW 0048 WD       0        0 S..   X  -   SyncInProg
0061 RW       0        0 RW 0049 WD       0        0 S..   X  -   Synchronized
0062 RW       0        0 RW 004A WD       0        0 S..   X  -   Synchronized
0063 RW       0        0 RW 004B WD       0        0 S..   X  -   Synchronized
0064 RW       0        0 RW 004C WD       0        0 S..   X  -   Synchronized
0065 RW       0        0 RW 004D WD       0        0 S..   X  -   Synchronized

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                LI       ST                       C  S
      A                 N        A                       o  u
      T  R1 Inv  R2 Inv K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks  S Dev    E  Tracks  Tracks MDA   s  p   STATE
```

```
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     1968 RW 0045 WD       0       0 S..    X  -   SyncInProg
005E RW       0     2785 RW 0046 WD       0       0 S..    X  -   SyncInProg
005F RW       0     2766 RW 0047 WD       0       0 S..    X  -   SyncInProg
0060 RW       0     2513 RW 0048 WD       0       0 S..    X  -   SyncInProg
0061 RW       0     2500 RW 0049 WD       0       0 S..    X  -   SyncInProg
0062 RW       0      426 RW 004A WD       0       0 S..    X  -   SyncInProg
0063 RW       0      430 RW 004B WD       0       0 S..    X  -   SyncInProg
0064 RW       0        0 RW 004C WD       0       0 S..    X  -   Synchronized
0065 RW       0        0 RW 004D WD       0       0 S..    X  -   Synchronized

Total   ------- -------            ------- -------
Trks          0   28582                  0       0
MBs         0.0   893.0                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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

Example 3: Tripping a Consistency Group Manually

This example continues from the end of Example 2 to determine if tripping the consistency group manually produces similar results to an automatic trip of the consistency group. This example also requires that I/O be occurring on both Symmetrix arrays.

The `symrdf verify` command checks the state of the SRDF pairs in the consistency group every five seconds until the pairs are synchronized. This ensures that the establish operation from the previous example is complete.

```
symrdf -cg oracle -synchronized verify -i 5

All devices in the CG group 'oracle' are in the 'Synchronized' state.
```

- The `symrdf suspend` command manually trips the consistency group. The `-force` option is required here to ensure that you really want to stop the SRDF mirroring operation and end consistency protection.

```
symrdf -cg oracle suspend -noprompt -force

An RDF 'Suspend' operation execution is in progress for composite group
   'oracle'.
   Please wait...

     Pend I/O on RDF link(s) for device(s) in (3087,01).............Done.
     Pend I/O on RDF link(s) for device(s) in (3143,01).............Done.
     Suspend RDF link(s) for device(s) in (3087,01).................Done.
     Suspend RDF link(s) for device(s) in (3143,01).................Done.

The RDF 'Suspend' operation successfully executed for composite group
   'oracle'.
```

- You can use the `symrdf verify` command with the `-susp_offline` option to verify that all R1 devices from the consistency group are in the Suspended Offline state. The following command checks every five seconds and, in this case, indicates that the trip has completed.

```
symrdf -cg oracle verify -susp_offline -i 5

All devices in the CG group 'oracle' are in both the 'Suspended' rdf state
   and the 'Offline' link suspend state.
```

- The following query confirms the state of the consistency group.

```
symrdf -cg oracle query

Composite Group Name      :  oracle
Composite Group Type      :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI      ST                      C  S
      A                   N       A                      o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     186 NR 0045 WD       0       0 S..   X  .   Suspended
005E RW       0     186 NR 0046 WD       0       0 S..   X  .   Suspended
005F RW       0       1 NR 0047 WD       0       0 S..   X  .   Suspended
0060 RW       0       0 NR 0048 WD       0       0 S..   X  .   Suspended
0061 RW       0       0 NR 0049 WD       0       0 S..   X  .   Suspended
0062 RW       0       0 NR 004A WD       0       0 S..   X  .   Suspended
0063 RW       0       0 NR 004B WD       0       0 S..   X  .   Suspended
0064 RW       0       0 NR 004C WD       0       0 S..   X  .   Suspended
0065 RW       0       0 NR 004D WD       0       0 S..   X  .   Suspended

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI      ST                      C  S
      A                   N       A                      o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
005D RW       0     367 NR 0045 WD       0       0 S..   X  .   Suspended
005E RW       0     300 NR 0046 WD       0       0 S..   X  .   Suspended
005F RW       0     300 NR 0047 WD       0       0 S..   X  .   Suspended
0060 RW       0    1343 NR 0048 WD       0       0 S..   X  .   Suspended
0061 RW       0    1330 NR 0049 WD       0       0 S..   X  .   Suspended
0062 RW       0      86 NR 004A WD       0       0 S..   X  .   Suspended
0063 RW       0      75 NR 004B WD       0       0 S..   X  .   Suspended
0064 RW       0       0 NR 004C WD       0       0 S..   X  .   Suspended
0065 RW       0       0 NR 004D WD       0       0 S..   X  .   Suspended

Total    ------- -------             ------- -------
Trks           0    4174                   0       0
MBs          0.0   130.0                 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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf resume` command resumes the RDF links between the SRDF pairs in the consistency group and I/O traffic between the R1 devices and their paired R2 devices. Normal SRDF mirroring resumes.

  `symrdf -cg oracle resume -noprompt`

```
An RDF 'Resume' operation execution is in progress for composite group
   'oracle'.
   Please wait...

     Resume RDF link(s) for device(s) in (3087,01)...................Done.
     Resume RDF link(s) for device(s) in (3143,01)...................Done.

The RDF 'Resume' operation successfully executed for composite group
   'oracle'.
```

- The `symrdf query` command displays the state of the SRDF pairs. Some pairs are in the process of synchronizing (SyncInProg), while others have completed synchronizing (Synchronized). The ellipsis (...) at the end indicates where the "Legend" output was omitted for brevity.

`symrdf -cg oracle query`

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID         : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
----------------------- ------------------------ ----- ------ ------------
     ST                  LI      ST                    C  S
      A                   N       A                    o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv    n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ---------------------- ----- ------ ------------
005D RW       0        7 RW 0045 WD       0        0 S..   X  -   SyncInProg
005E RW       0       12 RW 0046 WD       0        0 S..   X  -   SyncInProg
005F RW       0        0 RW 0047 WD       0        0 S..   X  -   Synchronized
0060 RW       0        0 RW 0048 WD       0        0 S..   X  -   Synchronized
0061 RW       0        0 RW 0049 WD       0        0 S..   X  -   Synchronized
0062 RW       0        0 RW 004A WD       0        0 S..   X  -   Synchronized
0063 RW       0        0 RW 004B WD       0        0 S..   X  -   Synchronized
0064 RW       0        0 RW 004C WD       0        0 S..   X  -   Synchronized
0065 RW       0        0 RW 004D WD       0        0 S..   X  -   Synchronized

Symmetrix ID                : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View        Target (R2) View       MODES STATES
----------------------- ------------------------ ----- ------ ------------
     ST                  LI      ST                    C  S
      A                   N       A                    o  u
      T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv    n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ---------------------- ----- ------ ------------
005D RW       0      221 RW 0045 WD       0        0 S..   X  -   SyncInProg
005E RW       0      110 RW 0046 WD       0        0 S..   X  -   SyncInProg
005F RW       0      130 RW 0047 WD       0        0 S..   X  -   SyncInProg
0060 RW       0      640 RW 0048 WD       0        0 S..   X  -   SyncInProg
0061 RW       0      626 RW 0049 WD       0        0 S..   X  -   SyncInProg
0062 RW       0       34 RW 004A WD       0        0 S..   X  -   SyncInProg
```

```
0063 RW       0      44 RW 004B WD       0       0 S..   X  -   SyncInProg
0064 RW       0       0 RW 004C WD       0       0 S..   X  -   Synchronized
0065 RW       0       0 RW 004D WD       0       0 S..   X  -   Synchronized

Total   ------- -------            ------- -------
Trks          0    1824                  0       0
MBs         0.0    57.0                0.0     0.0
```

Example 4: Creating a Composite Group from Existing Sources

If you have existing sources that define devices that you want to include in your composite group, you can translate one of these sources into a new composite group rather than build the composite group as described in Example 1. This example populates a composite group using existing devices from different sources: a device group, an RDBMS database, an RDBMS tablespace, and a logical volume group.

Creating a Composite Group from a Device Group

The following commands were issued from a Solaris host (sol218).

- The `symdg list` command displays two device groups (ora1 and ora2) containing devices that can be included in a composite group.

`symdg list`

```
                          D E V I C E     G R O U P S

                                                         Num of    Num of   Num of
  Name                 Type      Valid  Symmetrix ID     Devices   GK's     BCV's

  ora1                  RDF1      Yes     000000003087       9          0        0
  ora2                  RDF1      Yes     000000003143       9          0        0
```

- The `symdg dg2cg` command creates and populates a composite group named oracle, using devices from a device group named ora1. The `-ppath` option creates the composite group in the host's PowerPath database.

  `symdg dg2cg ora1 oracle -ppath`

```
Adding STD device 005D to group 'oracle'... OK
Adding STD device 005E to group 'oracle'... OK
Adding STD device 005F to group 'oracle'... OK
Adding STD device 0060 to group 'oracle'... OK
Adding STD device 0061 to group 'oracle'... OK
Adding STD device 0062 to group 'oracle'... OK
Adding STD device 0063 to group 'oracle'... OK
Adding STD device 0064 to group 'oracle'... OK
Adding STD device 0065 to group 'oracle'... OK

9 device(s) were added to group 'oracle'.
```

- The `symcg list` command displays a list of composite groups defined on this host. This display shows the new composite group and that nine devices were added to it from the device group.

`symcg list`

```
                      C O M P O S I T E   G R O U P S

                                             Number of          Number of
  Name                       Type     Valid  Symms    RAGs     Devs    BCVs     VDEVs

  oracle                     RDF1     Yes        1       1        9       0         0
```

- This `symdg dg2cg` command adds more devices to the same composite group. These devices are from the device group named ora2.

  `symdg dg2cg ora2 oracle -ppath`

  ```
  Adding STD device 005D to group 'oracle'... OK
  Adding STD device 005E to group 'oracle'... OK
  Adding STD device 005F to group 'oracle'... OK
  Adding STD device 0060 to group 'oracle'... OK
  Adding STD device 0061 to group 'oracle'... OK
  Adding STD device 0062 to group 'oracle'... OK
  Adding STD device 0063 to group 'oracle'... OK
  Adding STD device 0064 to group 'oracle'... OK
  Adding STD device 0065 to group 'oracle'... OK

  9 device(s) were added to group 'oracle'.
  ```

- This `symcg list` command shows that the oracle composite group now contains the eighteen devices, nine from each device group.

`symcg list`

```
                        C O M P O S I T E   G R O U P S

                                          Number of          Number of
 Name                    Type    Valid  Symms    RAGs    Devs    BCVs    VDEVs

 oracle                  RDF1    Yes        2       2      18       0        0
```

Creating a Composite Group from an RDBMS Database or Tablespace

The following commands were issued from a Solaris host (sol218).

For SYMCLI to access a specified database, set the `SYMCLI_RDB_CONNECT` environment variable to the username and password of the system administrator's account. The first `export` command sets this variable to a username of "system" and a password of "manager." It also specifies the Oracle connection string "ora217," a TNS-alias name that is required if connecting via the network instead of locally. The `export ORACLE_HOME` command specifies the location of the Oracle binaries. The `export ORACLE_SID` command specifies the database instance name. (Other RDBMS database systems use environment variables specific to their system. For example, Sybase uses the variable `SYBASE` for the location of the Sybase binaries and uses `DSQUERY` to specify the server name.)

```
export SYMCLI_RDB_CONNECT=system/manager@ora217
export ORACLE_HOME=/disks/symapidvt/oraclehome/ora217
export ORACLE_SID=ora217
```

- The `symcg list` command shows no composite groups currently defined on this host.

  `symcg list`

  ```
  No Symmetrix composite groups were found
  ```

- The `symrdb rdb2cg` command creates and populates an RDF1 type composite group named oraclecg from an oracle type database named ora217. The verbose option (`-v`) produces a detailed output display. The `-ppath` option creates the composite group in the host's PowerPath database.

`symrdb -type oracle -db ora217 -v rdb2cg oraclecg -cgtype rdf1 -ppath`

```
Physical Device: /dev/vx/rdmp/c1t0d39s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c1t0d50s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c1t0d51s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c2t0d49s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c2t0d50s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c1t0d52s2 CG ADD SUCCEEDED
6 devices were added to composite group 'oraclecg'.
```

- The `symcg list` command confirms the new composite group and its contents.

`symcg list`

```
                      C O M P O S I T E   G R O U P S

                                         Number of          Number of
 Name                   Type     Valid  Symms    RAGs     Devs    BCVs    VDEVs

 oraclecg               RDF1     Yes        2       2        6       0        0
```

- The `symrdb list tbs` command displays tablespace names in the oracle database ora217.

`symrdb list -type oracle tbs`

```
TABLE SPACE NAMES (ORACLE 8.1.7.0.0):

  Table Space  Name                 Type       Status
  -------------------               ------     ---------

  SYSTEM                            Permanent  Online
  RBS                               Permanent  Online
  TEMP                              Temporary  Online
  TPCB                              Permanent  Online
```

- The `symrdb tbs2cg` command creates and populates an RDF1 type composite group named oracletbscg from a tablespace named system. The tablespace is within the oracle type database named ora217 (from the environment variable setting). The `-ppath` option creates the composite group in the host's PowerPath database.

`symrdb -type oracle -tbs system -v  tbs2cg oracletbscg -cgtype rdf1 -ppath`

```
Physical Device: /dev/vx/rdmp/c2t0d48s2 CG ADD SUCCEEDED
Physical Device: /dev/vx/rdmp/c2t0d49s2 CG ADD SUCCEEDED
2 devices were added to composite group 'oracletbscg'.
```

- The `symcg list` command confirms the new composite group and its contents. Note that the previous composite group (oraclecg) was deleted before creating this new group.

`symcg list`

```
                      C O M P O S I T E   G R O U P S

                                         Number of          Number of
 Name                   Type     Valid  Symms    RAGs     Devs    BCVs    VDEVs

 oracletbscg            RDF1     Yes        1       1        2       0        0
```

Creating a Composite Group from a Logical Volume Group

The following commands were issued from a HP-UX host (hpu106).

- The `symvg list` command displays the logical volume groups that have been defined for this host.

```
symvg list

VOLUME GROUPS (HP-UX LVM):

                                              PE     Max     Max     Num     Num
  Name            State     Attribute       Size  Devices Volumes Devices Volumes

  /dev/vg00        Enabled  N/A            4096k       16     255       1       8
  /dev/orax        Enabled  N/A            4096k       16     255       3       0
```

- The `symcg list` command determines that there are currently no composite groups defined on this host.

```
symcg list

No Symmetrix composite groups were found
```

- The `symvg vg2cg` command creates and populates an RDF1 type composite group named lvmcg from the volume group named orax. The `-ppath` option creates the composite group in the host's PowerPath database.

```
symvg vg2cg orax lvmcg -cgtype rdf1 -ppath

3 device(s) were added to group 'lvmcg'.
```

- The `symcg list` command confirms the new composite group and its contents.

```
symcg list

                        C O M P O S I T E   G R O U P S

                                            Number of          Number of
    Name                     Type     Valid  Symms    RAGs     Devs    BCVs    VDEVs

    lvmcg                    RDF1     Yes        1       1        3       0        0
```

Example 5: A CG that Spans Two Hosts Writing to Two Symmetrix Arrays

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host (sol218) and an HP-UX host (hpu106), each connected to the same two local Symmetrix arrays (Symmetrix 3087 and Symmetrix 3143). Each of these local Symmetrix arrays is connected via RDF links to a remote Symmetrix array (Symmetrix 3003 and Symmetrix 3156, respectively).

The two hosts write to different PowerPath devices on the local Symmetrix arrays. Each host has the same consistency group named "oracle" defined on it, but each host has access only to the devices that hold its own database. If both hosts are writing to their respective R1 devices and one of those R1s cannot propagate data to its R2 device, the consistency group is tripped. This means each R2-side host connected to the remote Symmetrix arrays can start a copy of its respective R2 database that is consistent with data on the R1 devices up to the time of the trip.

PowerPath 2.1.1 software and Oracle 8.1.7.0.0 software are installed on each host. An Oracle database has been created on each host.

The consistency group called oracle contains the following sixteen devices:

- PowerPath devices 0001, 0002, 0003, 0004 on Symmetrix 3087 (visible only to Solaris host sol218)
- PowerPath devices 010C, 010D, 010E, 010F on Symmetrix 3143 (visible only to Solaris host sol218)
- PowerPath devices 0070, 008F, 0090, 0091 on Symmetrix 3087 (visible only to HP-UX host hpu106)
- PowerPath devices 006E, 006F, 0070, 0071 on Symmetrix 3143 (visible only to HP-UX host hpu106)

Figure 7-1 illustrates this configuration. Host sol218 writes to four devices on Symmetrix 3087 (0001-0004) and to four devices on Symmetrix 3143 (0070-0091). Host hpu106 writes to four devices on Symmetrix 3087 (0070, 008F, 0090, and 0091) and to four devices on Symmetrix 3143 (006E-0071). All sixteen devices belong to the same consistency group. The eight devices that sol218 writes to hold one database. The eight devices that hpu106 writes to hold another database.

If any of the R1 devices cannot propagate data to its R2 device, the consistency group is automatically tripped. The result is two consistent, DBMS-restartable copies of the databases on Symmetrix 3003 and Symmetrix 3156.

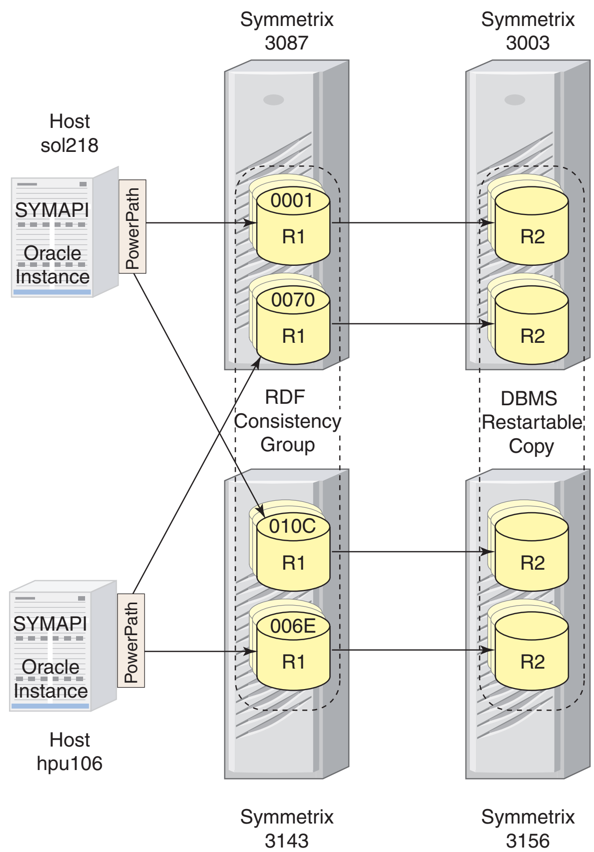

Figure 7-1 Using a Consistency Group that Spans Two Hosts Writing to Two Symmetrix Arrays

- The `symcfg list` command displays high-level information on Symmetrix arrays available to host hpu106. The Num Symm Devices column displays the total number of Symmetrix devices configured for a Symmetrix array. The Num Phys Devices column displays only devices that are mapped to host hpu106 and/or have a physical host device name. Host hpu106 has access (read/write capability) to 204 and 78 physical devices on each of the local Symmetrix arrays, respectively. But the local host (or point-of-view host in this case) cannot access devices on remote Symmetrix arrays, which accounts for the zero values under Num Phys Devices.

`symcfg list`

```
                                S Y M M E T R I X

                                         Mcode    Cache      Num Phys  Num Symm
    SymmID       Attachment  Model      Version  Size (MB)  Devices   Devices

    000000003087 Local        DMX800     5670       6144      204       812
    000000003143 Local        DMX800     5670       6144       78       526
    000000003003 Remote       DMX800     5670       6144        0       812
    000000003156 Remote       DMX800     5670       6144        0       526
```

- The `sympd list` command from host hpu106 displays the devices on each local Symmetrix array that are visible to this host. The ellipsis (......) represents omitted or truncated output.

sympd list

```
Symmetrix ID: 000000003087

       Device Name           Directors                  Device
---------------------------- ------------ ----------------------------------
                                                                       Cap
Physical                   Sym   SA :P DA :IT Config       Attribute    Sts   (MB)
---------------------------- ------------ ----------------------------------

/dev/rdsk/c0t0d0         001C 15B:0 02A:D1 Unprotected   N/Grp'd      RW
  4315
......................................................................................
/dev/rdsk/c0t1d7         0070 15B:0 02B:C2 RDF1+Mir      N/Grp'd  (M) RW 12946
/dev/rdsk/c0t2d0        008F 15B:0 01A:D2 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d1        0090 15B:0 02B:C2 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d2        0091 15B:0 01B:C2 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d3        0092 15B:0 02A:C2 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d4        0093 15B:0 01A:C2 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d5        0094 15B:0 02B:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d6        0095 15B:0 01B:D1 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t2d7        00B2 15B:0 02A:C2 2-Way Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t3d0        00B3 15B:0 01A:C2 2-Way Mir     N/Grp'd      RW    4315
......................................................................................

Symmetrix ID: 000000003143

       Device Name           Directors                  Device
---------------------------- ------------ ----------------------------------
                                                                       Cap
Physical                   Sym   SA :P DA :IT Config       Attribute    Sts   (MB)
---------------------------- ------------ ----------------------------------

/dev/rdsk/c1t0d0        000C 15B:0 02B:C2 2-Way Mir     N/Grp'd      RW    4315
......................................................................................
/dev/rdsk/c1t3d6        006E 15B:0 02A:D3 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c1t3d7        006F 15B:0 01A:D3 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c1t4d0        0070 15B:0 02B:C3 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c1t4d1        0071 15B:0 01B:C3 RDF1+Mir      N/Grp'd      RW    4315
/dev/rdsk/c1t4d2         009C 15B:0 02B:D3 RDF1+Mir      N/Grp'd  (M) RW 17261
/dev/rdsk/c1t4d3         00AC 15B:0 02B:D1 RDF1+Mir      N/Grp'd  (M) RW 17261
/dev/rdsk/c1t6d0        008C 15B:0 02B:C1 Unprotected   N/Grp'd      RW       3
......................................................................................
```

- From host hpu106, you can use `sympd list` with the `-powerpath` option to display just the PowerPath devices on the local Symmetrix arrays.

sympd list -powerpath

```
Symmetrix ID: 000000003087

                    P O W E R P A T H     D E V I C E S

       Device Name           Directors                  Device
---------------------------- ------------ ----------------------------------
                                                                       Cap
Physical                   Sym   SA :P DA :IT Config       Attribute    Sts   (MB)
---------------------------- ------------ ----------------------------------
                          001C        02A:D1 Unprotected  N/Grp'd      RW    4315
/dev/rdsk/c0t0d0              - 15B:0      - -           -            -        -
......................................................................................
                          0070        02B:C2 RDF1+Mir     N/Grp'd  (M) RW   12946
```

```
/dev/rdsk/c0t1d7          - 15B:0      - -         -            -       -
                          008F      01A:D2 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d0          - 15B:0      - -         -            -       -
                          0090      02B:C2 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d1          - 15B:0      - -         -            -       -
                          0091      01B:C2 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d2          - 15B:0      - -         -            -       -
                          0092      02A:C2 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d3          - 15B:0      - -         -            -       -
                          0093      01A:C2 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d4          - 15B:0      - -         -            -       -
                          0094      02B:D1 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d5          - 15B:0      - -         -            -       -
                          0095      01B:D1 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c0t2d6          - 15B:0      - -         -            -       -
                          00B2      02A:C2 2-Way Mir    N/Grp'd      RW    4315
.................................................................................
Symmetrix ID: 000000003143
                      P O W E R P A T H     D E V I C E S
       Device Name             Directors              Device
--------------------------- ------------ ---------------------------------
                                                                      Cap
Physical                   Sym  SA :P DA :IT Config       Attribute    Sts   (MB)
--------------------------- ------------ ---------------------------------
                          000C      02B:C2 2-Way Mir    N/Grp'd      RW    4315
/dev/rdsk/c1t0d0          - 15B:0      - -         -            -       -
.................................................................................
                          006E      02A:D3 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c1t3d6          - 15B:0      - -         -            -       -
                          006F      01A:D3 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c1t3d7          - 15B:0      - -         -            -       -
                          0070      02B:C3 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c1t4d0          - 15B:0      - -         -            -       -
                          0071      01B:C3 RDF1+Mir     N/Grp'd      RW    4315
/dev/rdsk/c1t4d1          - 15B:0      - -         -            -       -
                          009C      02B:D3 RDF1+Mir     N/Grp'd  (M) RW   17261
/dev/rdsk/c1t4d2          - 15B:0      - -         -            -       -
                          00AC      02B:D1 RDF1+Mir     N/Grp'd  (M) RW   17261
/dev/rdsk/c1t4d3          - 15B:0      - -         -            -       -
                          008C      02B:C1 Unprotected  N/Grp'd      RW       3
.................................................................................
```

- The `symcg create` command from host hpu106 creates a composite group name oracle on this host. Beginning with Solutions Enabler version 5.4, you must specify an RDF type to make the group capable of being enabled for consistency protection. Moreover, if you have not already set the SYMAPI_RDF_CG_TO_PPATH option to ENABLE, you must include the `–ppath` option so that the group is added to PowerPath.

  `symcg create oracle -type rdf1 -ppath`

- The `symcg list` command from host hpu106 displays composite groups defined on this host. Only one composite group is defined, and it does not yet contain any devices. Beginning with Solutions Enabler version 5.4, you can include the `–ppath` option to display only those groups that are in PowerPath. When the composite group is put into PowerPath, the Type remains N/A until the first standard device is added to the group.

```
symcg list -ppath

                     C O M P O S I T E   G R O U P S

                                          Number of          Number of
    Name                   Type    Valid  Symms    RAGs    Devs    BCVs    VDEVs

    oracle                 N/A     Yes        0       0       0       0        0

The symcg add commands add eight devices from Symmetrix 3087 and eight from
   Symmetrix 3143 to the composite group named oracle.
symcg -cg oracle -sid 3087 add dev 0001
symcg -cg oracle -sid 3087 add dev 0002
symcg -cg oracle -sid 3087 add dev 0003
symcg -cg oracle -sid 3087 add dev 0004
symcg -cg oracle -sid 3087 add dev 0070
symcg -cg oracle -sid 3087 add dev 008F
symcg -cg oracle -sid 3087 add dev 0090
symcg -cg oracle -sid 3087 add dev 0091
symcg -cg oracle -sid 3143 add dev 010C
symcg -cg oracle -sid 3143 add dev 010D
symcg -cg oracle -sid 3143 add dev 010E
symcg -cg oracle -sid 3143 add dev 010F
symcg -cg oracle -sid 3143 add dev 006E
symcg -cg oracle -sid 3143 add dev 006F
symcg -cg oracle -sid 3143 add dev 0070
symcg -cg oracle -sid 3143 add dev 0071
```

- The `symcg list` command from host hpu106 shows that there are now sixteen devices in this composite group.

```
symcg list

                     C O M P O S I T E   G R O U P S

                                          Number of          Number of
    Name                   Type    Valid  Symms    RAGs    Devs    BCVs    VDEVs

    oracle                 RDF1    Yes        2       2      16       0        0
```

- The `symcg show` command from hpu106 displays detailed configuration and status information about the composite group. Note that the current Consistency State of the devices is Disabled.

`symcg show oracle`

```
Composite Group Name:  oracle

  Composite Group Type                                : RDF1
  Valid                                                : Yes
  CG in PowerPath                                      : Yes
  CG in GNS                                            : No

  Number of RDF (RA) Groups                            :    2
  Number of STD Devices                                :   16
  Number of BCV's (Locally-associated)                 :    0
  Number of VDEV's (Locally-associated)                :    0
  Number of RBCV's (Remotely-associated STD-RDF)       :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)      :    0
  Number of RR BCV's (Remotely-associated RBCV)        :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                     : 000000003087
        Microcode Version                                : 5670
        Number of STD Devices                            :    8
        Number of BCV's (Locally-associated)             :    0
        Number of VDEV's (Locally-associated)            :    0
        Number of RBCV's (Remotely-associated STD_RDF)   :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):     0
        Number of RRBCV's (Remotely-associated RBCV)     :    0

        1) RDF (RA) Group Number   :  1             (00)
           Remote Symmetrix ID     : 000000003003

           Symmetrix Devices (8):
              {
            ------------------------------------------------------------
                                       Sym  Device        Consistency  Cap
             PdevName                  Dev  Config        State        (MB)
            ------------------------------------------------------------
             /dev/rdsk/c0t1d7          0070 RDF1+Mir      Disabled     12946
             /dev/rdsk/c0t2d0          008F RDF1+Mir      Disabled      4315
             /dev/rdsk/c0t2d1          0090 RDF1+Mir      Disabled      4315
             /dev/rdsk/c0t2d2          0091 RDF1+Mir      Disabled      4315
             N/A                       0001 RDF1          Disabled      4315
             N/A                       0002 RDF1          Disabled      4315
             N/A                       0003 RDF1          Disabled      4315
             N/A                       0004 RDF1          Disabled      4315
              }
        }

     2) Symmetrix ID                                     : 000000003143
        Microcode Version                                : 5670
        Number of STD Devices                            :    8
        Number of BCV's (Locally-associated)             :    0
        Number of VDEV's (Locally-associated)            :    0
        Number of RBCV's (Remotely-associated STD_RDF)   :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):     0
        Number of RRBCV's (Remotely-associated RBCV)     :    0

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number   :  1             (00)
```

```
            Remote Symmetrix ID    : 000000003156
            Symmetrix Devices (8):
              {
             -------------------------------------------------------------
                                        Sym   Device        Consistency   Cap
              PdevName                  Dev   Config        State         (MB)
             -------------------------------------------------------------
              /dev/rdsk/c1t3d6          006E RDF1+Mir       Disabled      4315
              /dev/rdsk/c1t3d7          006F RDF1+Mir       Disabled      4315
              /dev/rdsk/c1t4d0          0070 RDF1+Mir       Disabled      4315
              /dev/rdsk/c1t4d1          0071 RDF1+Mir       Disabled      4315
              N/A                       010C RDF1           Disabled      4315
              N/A                       010D RDF1           Disabled      4315
              N/A                       010E RDF1           Disabled      4315
              N/A                       010F RDF1           Disabled      4315
              }
          }
    }
```

- Because the same composite group definition must exist on both local hosts, it is necessary to build[1] a composite group on sol218 that has the same definitions as the composite group on hpu106. The `symcg export` command from host hpu106 creates a text file (`oracle_at_106.cg`) that contains the composite group definitions. You then use `rcp` (or `ftp`) to transfer that file to host sol218.

```
symcg -f oracle_at_106.cg export oracle
rcp  oracle_at_106.cg sol218:/.
```

- From host sol218, issue the `symcg import` command to build a composite group called oracle on sol218, using the definitions from the text file. Although oracle is the name used for the composite group on both hosts, the names for the two groups do not have to be the same. Only the content must be the same.

```
symcg -f oracle_at_106.cg import oracle -ppath

Adding device 0070 on Symmetrix 000000003087 to 'oracle'...
Adding device 008F on Symmetrix 000000003087 to 'oracle'...
Adding device 0090 on Symmetrix 000000003087 to 'oracle'...
Adding device 0091 on Symmetrix 000000003087 to 'oracle'...
Adding device 0001 on Symmetrix 000000003087 to 'oracle'...
Adding device 0002 on Symmetrix 000000003087 to 'oracle'...
Adding device 0003 on Symmetrix 000000003087 to 'oracle'...
Adding device 0004 on Symmetrix 000000003087 to 'oracle'...
Adding device 006E on Symmetrix 000000003143 to 'oracle'...
Adding device 006F on Symmetrix 000000003143 to 'oracle'...
Adding device 0070 on Symmetrix 000000003143 to 'oracle'...
Adding device 0071 on Symmetrix 000000003143 to 'oracle'...
Adding device 010C on Symmetrix 000000003143 to 'oracle'...
Adding device 010D on Symmetrix 000000003143 to 'oracle'...
Adding device 010E on Symmetrix 000000003143 to 'oracle'...
Adding device 010F on Symmetrix 000000003143 to 'oracle'...
```

1. Beginning with Solutions Enabler version 5.4, if Group Name Services (GNS) is enabled and the GNS daemon is running on the local hosts, the composite group definition is *automatically* propagated to the local Symmetrix arrays and to other hosts connected to these arrays. For details, refer to the *EMC Solutions Enabler Symmetrix Array Management Product Guide*.

◆ The `symcg list` command issued from sol218 displays the composite group as it is now defined on host sol218.

```
symcg list

                        C O M P O S I T E   G R O U P S

                                                 Number of          Number of
    Name                     Type    Valid   Symms    RAGs    Devs    BCVs    VDEVs

    oracle                   RDF1    Yes         2       2      16       0        0
```

◆ The `symcg enable` command from host sol218 enables consistency protection for device pairs in the composite group. The group is now known as a *consistency group*.

```
symcg -cg oracle enable -noprompt

A consistency 'Enable' operation execution is in progress for composite group
   'oracle'. Please wait...

The composite group 'Enable' operation successfully executed for composite
   group
   'oracle'.
```

◆ The `symcg show` command issued from sol218 displays details of the consistency group from sol218's point of view. Note that devices that are not visible (N/A under PdevName) to sol218 are the devices that are visible to hpu106, and vice versa. The consistency state is now enabled.

```
symcg show oracle

Composite Group Name:  oracle

  Composite Group Type                                  : RDF1
  Valid                                                  : Yes
  CG in PowerPath                                        : Yes
  CG in GNS                                              : No

  Number of RDF (RA) Groups                              :    2
  Number of STD Devices                                  :   16
  Number of BCV's (Locally-associated)                   :    0
  Number of VDEV's (Locally-associated)                  :    0
  Number of RBCV's (Remotely-associated STD-RDF)         :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)        :    0
  Number of RR BCV's (Remotely-associated RBCV)          :    0

  Number of Symmetrix Units (2):
     {
     1) Symmetrix ID                                       : 000000003087
        Microcode Version                                  : 5670
        Number of STD Devices                              :    8
        Number of BCV's (Locally-associated)               :    0
        Number of VDEV's (Locally-associated)              :    0
        Number of RBCV's (Remotely-associated STD_RDF)    :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)      :    0

        Number of RDF (RA) Groups (1):
           {
           1) RDF (RA) Group Number    :  1             (00)
              Remote Symmetrix ID      : 000000003003

              Symmetrix Devices (8):
                 {
```

```
                 --------------------------------------------------------------
                                          Sym  Device        Consistency   Cap
                  PdevName                Dev  Config        State         (MB)
                 --------------------------------------------------------------
                  /dev/rdsk/emcpower25c   0001 RDF1          Enabled       4315
                  /dev/rdsk/emcpower26c   0002 RDF1          Enabled       4315
                  /dev/rdsk/emcpower27c   0003 RDF1          Enabled       4315
                  /dev/rdsk/emcpower28c   0004 RDF1          Enabled       4315
                  N/A                     0070 RDF1+Mir      Enabled      12946
                  N/A                     008F RDF1+Mir      Enabled       4315
                  N/A                     0090 RDF1+Mir      Enabled       4315
                  N/A                     0091 RDF1+Mir      Enabled       4315
                   }
             }

      2) Symmetrix ID                                   : 000000003143
         Microcode Version                              : 5670
         Number of STD Devices                          :    8
         Number of BCV's (Locally-associated)           :    0
         Number of VDEV's (Locally-associated)          :    0
         Number of RBCV's (Remotely-associated STD_RDF) :    0
         Number of BRBCV's (Remotely-associated BCV-RDF):    0
         Number of RRBCV's (Remotely-associated RBCV)   :    0

         Number of RDF (RA) Groups (1):
            {

            1) RDF (RA) Group Number  :  1             (00)
               Remote Symmetrix ID    : 000000003156

               Symmetrix Devices (8):
                  {
                 --------------------------------------------------------------
                                          Sym  Device        Consistency   Cap
                  PdevName                Dev  Config        State         (MB)
                 --------------------------------------------------------------
                  /dev/rdsk/emcpower0c    010C RDF1          Enabled       4315
                  /dev/rdsk/emcpower1c    010D RDF1          Enabled       4315
                  /dev/rdsk/emcpower2c    010E RDF1          Enabled       4315
                  /dev/rdsk/emcpower3c    010F RDF1          Enabled       4315
                  N/A                     006E RDF1+Mir      Enabled       4315
                  N/A                     006F RDF1+Mir      Enabled       4315
                  N/A                     0070 RDF1+Mir      Enabled       4315
                  N/A                     0071 RDF1+Mir      Enabled       4315
                   }
            }
      }
```

- The example switches back to hpu106 to perform the following SRDF control commands, although having the same consistency group defined on both hosts allows commands from either host. The `symcg show` command issued from hpu106 displays details of the consistency group from hpu106's point of view.

```
symcg show oracle

Composite Group Name:  oracle

  Composite Group Type                                  : RDF1
  Valid                                                  : Yes
  CG in PowerPath                                        : Yes
  CG in GNS                                              : No

  Number of RDF (RA) Groups                              :    2
  Number of STD Devices                                  :   16
  Number of BCV's (Locally-associated)                   :    0
  Number of VDEV's (Locally-associated)                  :    0
  Number of RBCV's (Remotely-associated STD-RDF)         :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)        :    0
  Number of RR BCV's (Remotely-associated RBCV)          :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                        : 000000003087
        Microcode Version                                   : 5670
        Number of STD Devices                               :    8
        Number of BCV's (Locally-associated)                :    0
        Number of VDEV's (Locally-associated)               :    0
        Number of RBCV's (Remotely-associated STD_RDF) :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)   :    0

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number  :  1            (00)
              Remote Symmetrix ID    : 000000003003
              Symmetrix Devices (8):
                 {
               -------------------------------------------------------------
                                           Sym  Device        Consistency  Cap
                PdevName                   Dev  Config        State        (MB)
               -------------------------------------------------------------
                /dev/rdsk/c0t1d7           0070 RDF1+Mir      Enabled     12946
                /dev/rdsk/c0t2d0           008F RDF1+Mir      Enabled      4315
                /dev/rdsk/c0t2d1           0090 RDF1+Mir      Enabled      4315
                /dev/rdsk/c0t2d2           0091 RDF1+Mir      Enabled      4315
                N/A                        0001 RDF1          Enabled      4315
                N/A                        0002 RDF1          Enabled      4315
                N/A                        0003 RDF1          Enabled      4315
                N/A                        0004 RDF1          Enabled      4315
                 }
           }

     2) Symmetrix ID                                        : 000000003143
        Microcode Version                                   : 5670
        Number of STD Devices                               :    8
        Number of BCV's (Locally-associated)                :    0
        Number of VDEV's (Locally-associated)               :    0
        Number of RBCV's (Remotely-associated STD_RDF) :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)   :    0

        Number of RDF (RA) Groups (1):
```

```
          {

          1) RDF (RA) Group Number  :  1            (00)
             Remote Symmetrix ID    : 000000003156

             Symmetrix Devices (8):
                {
              -------------------------------------------------------------
                                       Sym  Device        Consistency   Cap
               PdevName                Dev  Config        State         (MB)
              -------------------------------------------------------------
               /dev/rdsk/c1t3d6        006E RDF1+Mir      Enabled       4315
               /dev/rdsk/c1t3d7        006F RDF1+Mir      Enabled       4315
               /dev/rdsk/c1t4d0        0070 RDF1+Mir      Enabled       4315
               /dev/rdsk/c1t4d1        0071 RDF1+Mir      Enabled       4315
               N/A                     010C RDF1          Enabled       4315
               N/A                     010D RDF1          Enabled       4315
               N/A                     010E RDF1          Enabled       4315
               N/A                     010F RDF1          Enabled       4315
                }
          }
    }
```

- The `symrdf query` command from hpu106 checks the state of the SRDF pairs on both local Symmetrix arrays. Note that all devices in the consistency group are in both the RDF pair Split state and the offline link suspend state (indicated by a "." in the "Susp" column). Both the R1 and R2 devices can be accessed for read/write activity (RW) by their respective hosts. An "X" in the "Cons" column shows that the devices are enabled for consistency protection.

symrdf -cg oracle query

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                      C  S
      A                  N       A                      o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv      n  s    RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA  s  p    STATE
----------------------- -- ----------------------- ----- ------ ------------
0070 RW       0       0 NR 008A RW    3209       0 S..  X  .    Split
008F RW       0       0 NR 00A9 RW    3208       0 S..  X  .    Split
0090 RW       0       0 NR 00AA RW    3208       0 S..  X  .    Split
0091 RW       0       0 NR 00AB RW    3209       0 S..  X  .    Split
0001 RW       0       0 NR 001B RW    9626       0 S..  X  .    Split
0002 RW       0       0 NR 001C RW    9625       0 S..  X  .    Split
0003 RW       0       0 NR 001D RW    9625       0 S..  X  .    Split
0004 RW       0       0 NR 001E RW    9625       0 S..  X  .    Split

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                      C  S
      A                  N       A                      o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv      n  s    RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA  s  p    STATE
----------------------- -- ----------------------- ----- ------ ------------
006E RW       0       0 NR 0056 RW    3209       0 S..  X  .    Split
006F RW       0       0 NR 0057 RW    3208       0 S..  X  .    Split
0070 RW       0       0 NR 0058 RW    3208       0 S..  X  .    Split
0071 RW       0       0 NR 0059 RW    3208       0 S..  X  .    Split
010C RW       0       0 NR 010C RW    9625       0 S..  X  .    Split
010D RW       0       0 NR 010D RW    9625       0 S..  X  .    Split
010E RW       0       0 NR 010E RW    9626       0 S..  X  .    Split
010F RW       0       0 NR 010F RW    9625       0 S..  X  .    Split

Total   ------- -------           ------- -------
Trks          0       0            102669       0
MBs         0.0     0.0            3208.0     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

Legend for STATES:
```

```
Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf establish` command from host hpu106 initiates the synchronization of SRDF pairs in the consistency group. In the process, Enginuity compares the track tables of each source Symmetrix array and its target. If the track tables are not identical, the tables are merged. There are many reasons why track tables might change while the SRDF pairs are split, including new I/O to either the R1 side or R2 side.

`symrdf -cg oracle establish -noprompt`

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

    Write Disable device(s) in (3087,01) on RA at target (R2).......Done.
    Write Disable device(s) in (3143,01) on RA at target (R2).......Done.
    Suspend RDF link(s) for device(s) in (3087,01)..................Done.
    Suspend RDF link(s) for device(s) in (3143,01)..................Done.
   Mark target device(s) in (3087,01) to refresh from source.......Started.
   Device: 001B .................................................. Marked.
   Device: 001C .................................................. Marked.
   Device: 001D .................................................. Marked.
   Device: 001E .................................................. Marked.
   Devices: 008A-008C ............................................ Marked.
   Devices: 00A9-00AA ............................................ Marked.
   Device: 00AB .................................................. Marked.
    Mark target device(s) in (3087,01) to refresh from source.......Done.
   Mark target device(s) in (3143,01) to refresh from source.......Started.
   Devices: 0056-0057 ............................................ Marked.
   Devices: 0058-0059 ............................................ Marked.
   Device: 010C .................................................. Marked.
   Device: 010D .................................................. Marked.
   Device: 010E .................................................. Marked.
   Device: 010F .................................................. Marked.
    Mark target device(s) in (3143,01) to refresh from source.......Done.
    Suspend RDF link(s) for device(s) in (3087,01)..................Done.
   Merge track tables between source and target in (3087,01).......Started.
   Devices: 0001-0003 ............................................ Merged.
   Device: 0004 .................................................. Merged.
   Devices: 0070-0072 ............................................ Merged.
   Devices: 008F-0091 ............................................ Merged.
    Merge track tables between source and target in (3087,01).......Done.
    Suspend RDF link(s) for device(s) in (3143,01)..................Done.
   Merge track tables between source and target in (3143,01).......Started.
   Devices: 006E-0071 ............................................ Merged.
   Devices: 010C-010F ............................................ Merged.
    Merge track tables between source and target in (3143,01).......Done.
    Resume RDF link(s) for device(s) in (3087,01)...................Done.
    Resume RDF link(s) for device(s) in (3143,01)...................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   consistency
   group 'oracle'.
```

- The `symrdf query` command from host hpu106 displays the state of the SRDF pairs. All are in the process of synchronizing (state is SyncInProg).

symrdf -cg oracle query

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)

RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
      ST                  LI      ST                      C  S
       A                   N       A                      o  u
       T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s  RDF Pair
Dev    E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p  STATE
-----------------------  -- ----------------------- ----- ------ ------------
0070 RW       0    3209 RW 008A WD    3209       0 S..   X  .   SyncInProg
008F RW       0    3208 RW 00A9 WD    3208       0 S..   X  .   SyncInProg
0090 RW       0    3208 RW 00AA WD    3208       0 S..   X  .   SyncInProg
0091 RW       0    3209 RW 00AB WD    3209       0 S..   X  .   SyncInProg
0001 RW       0    9626 RW 001B WD    9626       0 S..   X  .   SyncInProg
0002 RW       0    9625 RW 001C WD    9625       0 S..   X  .   SyncInProg
0003 RW       0    9625 RW 001D WD    9625       0 S..   X  .   SyncInProg
0004 RW       0    9625 RW 001E WD    9625       0 S..   X  .   SyncInProg

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
      ST                  LI      ST                      C  S
       A                   N       A                      o  u
       T  R1 Inv  R2 Inv   K       T  R1 Inv  R2 Inv      n  s  RDF Pair
Dev    E  Tracks Tracks    S Dev   E  Tracks  Tracks MDA  s  p  STATE
-----------------------  -- ----------------------- ----- ------ ------------
006E RW       0    3209 RW 0056 WD    3209       0 S..   X  .   SyncInProg
006F RW       0    3208 RW 0057 WD    3208       0 S..   X  .   SyncInProg
0070 RW       0    1840 RW 0058 WD    1698       0 S..   X  .   SyncInProg
0071 RW       0    3209 RW 0059 WD    3208       0 S..   X  .   SyncInProg
010C RW       0    9625 RW 010C WD    9625       0 S..   X  .   SyncInProg
010D RW       0    9625 RW 010D WD    9625       0 S..   X  .   SyncInProg
010E RW       0    9626 RW 010E WD    9626       0 S..   X  .   SyncInProg
010F RW       0    9625 RW 010F WD    9625       0 S..   X  .   SyncInProg

Total   -------- --------           -------- -------
Trks           0        0            102669       0
MBs          0.0      0.0            3208.0     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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf verify` command checks the state of the SRDF pairs in the consistency group every 30 seconds until all pairs are in the Synchronized state.

  `symrdf -cg oracle verify -i 30 -synchronized`

```
NONE of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.
.............................................................................
NOT all of the mirrored pairs are in the 'Synchronized' state.

All devices in the CG group 'oracle' are in the 'Synchronized' state.
```

- The `symrdf split` command from hpu106 trips the consistency group, creating a DBMS-restartable copy of each database on the R2 target devices. The `-force` option is required here to ensure that you really want to stop the SRDF mirroring operation and end consistency protection.

  `symrdf -cg oracle split -noprompt -force`

```
An RDF 'Split' operation execution is in progress for composite group
   'oracle'.
   Please wait...

    Pend I/O on RDF link(s) for device(s) in (3087,01).............Done.
    Pend I/O on RDF link(s) for device(s) in (3143,01).............Done.
    Suspend RDF link(s) for device(s) in (3087,01).................Done.
    Suspend RDF link(s) for device(s) in (3143,01).................Done.
    Read/Write Enable device(s) in (3087,01) on RA at target (R2)...Done.
    Read/Write Enable device(s) in (3143,01) on RA at target (R2)...Done.

The RDF 'Split' operation successfully executed for composite group
   'oracle'.
```

- The following query from hpu106 confirms that all SRDF devices in the consistency group are in the Split state. It is now possible to start the two databases separately on two different R2-side hosts.

symrdf -cg oracle query

```
Composite Group Name       :  oracle
Composite Group Type       :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003087    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003003    (Microcode Version: 5670)

RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
-----------------------  -- ----------------------- ----- ------ ------------
0070 RW       0       0 NR 008A RW       0       0 S..   X  .   Split
008F RW       0       0 NR 00A9 RW       0       0 S..   X  .   Split
0090 RW       0       0 NR 00AA RW       0       0 S..   X  .   Split
0091 RW       0       0 NR 00AB RW       0       0 S..   X  .   Split
0001 RW       0       0 NR 001B RW       0       0 S..   X  .   Split
0002 RW       0       0 NR 001C RW       0       0 S..   X  .   Split
0003 RW       0       0 NR 001D RW       0       0 S..   X  .   Split
0004 RW       0       0 NR 001E RW       0       0 S..   X  .   Split

Symmetrix ID               : 000000003143    (Microcode Version: 5670)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5670)
RDF (RA) Group Number      :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                 LI      ST                       C  S
      A                  N       A                       o  u
      T  R1 Inv  R2 Inv  K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks   S Dev   E  Tracks  Tracks MDA   s  p   STATE
-----------------------  -- ----------------------- ----- ------ ------------
006E RW       0       0 NR 0056 RW       0       0 S..   X  .   Split
006F RW       0       0 NR 0057 RW       0       0 S..   X  .   Split
0070 RW       0       0 NR 0058 RW       0       0 S..   X  .   Split
0071 RW       0       0 NR 0059 RW       0       0 S..   X  .   Split
010C RW       0       0 NR 010C RW       0       0 S..   X  .   Split
010D RW       0       0 NR 010D RW       0       0 S..   X  .   Split
010E RW       0       0 NR 010E RW       0       0 S..   X  .   Split
010F RW       0       0 NR 010F RW       0       0 S..   X  .   Split

Total    ------- -------             ------- -------
Trks           0       0                   0       0
MBs          0.0     0.0                 0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

8

Performing SRDF/Automated Replication Operations

This chapter provides SYMCLI examples of specific actions and commands, which replicate data in pre-defined cycles using the SRDF automated replication process.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

Example 1: SRDF/AR Single-Hop Configuration

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a HP-UX host connected to a source Symmetrix (sid 161). All commands are issued from the source-side host. The example uses the following Symmetrix systems to create the single-hop environment:

- Local Source Symmetrix (sid 35): standard devices 56-6E; R1 BCV devices 182-19A
- Remote Target Symmetrix (sid 41): BCV devices 137-14F
- The `symcfg list` command displays the Symmetrix arrays attached to this host. Symmetrix arrays 000187900035 and 000187900041 are configured as a single-hop configuration.

`symcfg list`

```
                                 S Y M M E T R I X

                                          Mcode    Cache      Num Phys  Num Symm
 SymmID       Attachment  Model     Version  Size (MB)  Devices   Devices

 000000003143 Local       3630      5267       4096     209        447
 000000005232 Local       8230      5568       2048       6        652
 000184500160 Local       8430      5568      12288     132       2054
 000187700079 Local       DMX2000P  5670      51200     189       3326
 000187900035 Local       DMX800    5670       6144     204        812
 000000003156 Remote      3630      5267       4096       0        448
 000000005231 Remote      8230      5568       4096       0        714
 000000005233 Remote      8230      5568       6144       0        122
 000000006201 Remote      DMX2000P  5670      16384       0        824
 000184502898 Remote      8530      5568      12288       0       1139
 000187700067 Remote      2000P-M2  5670      51200       0       3762
 000187900041 Remote      DMX800    5670       6144       0        721
```

- The `symdg create` command creates a device group named symrep.

`symdg create symrep`

- The `sympd list` command displays all Symmetrix devices that are visible to this host. The display below has been edited to show those devices that are used in the example. The N/Grp'd attribute means that these devices are not already part of a device group and are free to be added to device group symrep.

```
sympd list -sid 35

Symmetrix ID: 000187900035

       Device Name          Directors                  Device
--------------------------- ------------- -----------------------------------
                                                                     Cap
Physical                    Sym  SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -----------------------------------

/dev/rdsk/c15t2d0s2      0008 01C:1 01A:CC  2-Way Mir     N/Grp'd       RW 3
/dev/rdsk/c15t2d1s2      0009 01C:1 16B:CC  2-Way Mir     N/Grp'd       RW 3
/dev/rdsk/c15t2d2s2      0056 01C:1 01B:CD  2-Way Mir     N/Grp'd  (M)  RW 12946
/dev/rdsk/c15t2d3s2      0059 01C:1 16A:C5  2-Way Mir     N/Grp'd  (M)  RW 8631
/dev/rdsk/c15t2d4s2      005B 01C:1 16B:C2  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d5s2      005C 01C:1 01A:CA  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d6s2      005D 01C:1 16A:C7  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d7s2      005E 01C:1 01B:C7  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d8s2      005F 01C:1 16B:C4  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d9s2      0060 01C:1 01A:C6  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d10s2     0061 01C:1 16A:C9  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d11s2     0062 01C:1 01B:C9  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d12s2     0063 01C:1 16B:C6  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d13s2     0064 01C:1 01A:C8  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d14s2     0065 01C:1 02A:CB  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d15s2     0066 01C:1 15B:CB  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d16s2     0067 01C:1 02B:CA  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d17s2     0068 01C:1 15A:CC  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d18s2     0069 01C:1 02A:CD  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d19s2     006A 01C:1 15B:C1  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d20s2     006B 01C:1 02B:C0  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d21s2     006C 01C:1 15A:CE  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d22s2     006D 01C:1 02A:C1  2-Way Mir     N/Grp'd       RW 4315
/dev/rdsk/c15t2d23s2     006E 01C:1 15B:C3  2-Way Mir     N/Grp'd       RW 4315
```

- The `symld` command adds one or more standard devices to the device group; the `-range` option can be used with the `addall` action to limit the selection to the devices that are within the specified range (for example, devices 56 through 6E).

  ```
  symld -g symrep addall dev -range 56:6E -sid 35
  ```

- The `symdev list` command with the `-r1 -bcv` options displays those R1 BCV devices in the local Symmetrix system (sid 35) that are not already part of a device group (N/Asst'd) and which are free to be added to the device group. The display below has been edited to show those devices that are used in the example.

```
symdev list -r1 -bcv -sid 35

Symmetrix ID: 000187900035

        Device Name           Directors                  Device
--------------------------- ------------- ------------------------------------
                                                                        Cap
Sym  Physical                SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- ------------------------------------

0182 Not Visible              ???:? 16A:CB  RDF1-BCV+Mir  N/Asst'd (M) NR 12946
0183 Not Visible             ???:? 01B:CB  RDF1-BCV+Mir  N/Asst'd (m)  NR      -
0184 Not Visible             ???:? 16B:CA  RDF1-BCV+Mir  N/Asst'd (m)  NR      -
0185 Not Visible              ???:? 01A:CC  RDF1-BCV+Mir  N/Asst'd (M) NR 8631
0186 Not Visible             ???:? 16A:CD  RDF1-BCV+Mir  N/Asst'd (m)  NR      -
0187 Not Visible             ???:? 01B:C1  RDF1-BCV+Mir  N/Asst'd      NR    4315
0188 Not Visible             ???:? 16B:C0  RDF1-BCV+Mir  N/Asst'd      NR    4315
0189 Not Visible             ???:? 01A:CE  RDF1-BCV+Mir  N/Asst'd      NR    4315
018A Not Visible             ???:? 16A:C1  RDF1-BCV+Mir  N/Asst'd      NR    4315
018B Not Visible             ???:? 01B:C3  RDF1-BCV+Mir  N/Asst'd      NR    4315
018C Not Visible             ???:? 16B:C8  RDF1-BCV+Mir  N/Asst'd      NR    4315
018D Not Visible             ???:? 01A:C0  RDF1-BCV+Mir  N/Asst'd      NR    4315
018E Not Visible             ???:? 16A:C3  RDF1-BCV+Mir  N/Asst'd      NR    4315
018F Not Visible             ???:? 01B:CD  RDF1-BCV+Mir  N/Asst'd      NR    4315
0190 Not Visible             ???:? 16B:CC  RDF1-BCV+Mir  N/Asst'd      NR    4315
0191 Not Visible             ???:? 01A:C2  RDF1-BCV+Mir  N/Asst'd      NR    4315
0192 Not Visible             ???:? 16A:C5  RDF1-BCV+Mir  N/Asst'd      NR    4315
0193 Not Visible             ???:? 01B:C5  RDF1-BCV+Mir  N/Asst'd      NR    4315
0194 Not Visible             ???:? 16B:CE  RDF1-BCV+Mir  N/Asst'd      NR    4315
0195 Not Visible             ???:? 01A:C4  RDF1-BCV+Mir  N/Asst'd      NR    4315
0196 Not Visible             ???:? 16A:C7  RDF1-BCV+Mir  N/Asst'd      NR    4315
0197 Not Visible             ???:? 01B:C7  RDF1-BCV+Mir  N/Asst'd      NR    4315
0198 Not Visible             ???:? 16B:C2  RDF1-BCV+Mir  N/Asst'd      NR    4315
0199 Not Visible             ???:? 01A:CA  RDF1-BCV+Mir  N/Asst'd      NR    4315
019A Not Visible             ???:? 16A:C9  RDF1-BCV+Mir  N/Asst'd      NR    4315
```

- The `symbcv` command associates the local BCV devices with the device group; the `-range` option is used with the `associate all` action to limit the selection to those BCVs that are within the specified range.

  `symbcv -g symrep associateall dev -range 182:19A`

- The `symdev list` command with the `-bcv` option displays those BCV devices in the remote Symmetrix (sid 41) that are not already part of a device group (N/Asst'd) and which are free to be added to the device group. The display below has been edited to show those devices that are used in the example.

`symdev list -sid 41 -bcv`

```
Symmetrix ID: 000187900041

        Device Name           Directors                  Device
---------------------------- ------------- -----------------------------------
                                                                         Cap
Sym  Physical                SA :P DA :IT  Config        Attribute    Sts   (MB)
---------------------------- ------------- -----------------------------------

0137 Not Visible             01C:0 02B:C8  BCV           N/Asst'd (M) RW   12946
0138 Not Visible             01C:0 01A:C8  BCV           N/Asst'd (m) RW       -
0139 Not Visible             01C:0 15A:C8  BCV           N/Asst'd (m) RW       -
013A Not Visible             01C:0 16A:CB  BCV           N/Asst'd (M) RW    8631
013B Not Visible             01C:0 02A:CB  BCV           N/Asst'd (m) RW       -
013C Not Visible             01C:0 01B:C7  BCV           N/Asst'd     RW    4315
013D Not Visible             01C:0 15B:C7  BCV           N/Asst'd     RW    4315
013E Not Visible             01C:0 16B:C0  BCV           N/Asst'd     RW    4315
013F Not Visible             01C:0 02B:C0  BCV           N/Asst'd     RW    4315
0140 Not Visible             01C:0 01A:C0  BCV           N/Asst'd     RW    4315
0141 Not Visible             01C:0 15A:C0  BCV           N/Asst'd     RW    4315
0142 Not Visible             01C:0 16A:CD  BCV           N/Asst'd     RW    4315
0143 Not Visible             01C:0 02A:CD  BCV           N/Asst'd     RW    4315
0144 Not Visible             01C:0 01B:CB  BCV           N/Asst'd     RW    4315
0145 Not Visible             01C:0 15B:CB  BCV           N/Asst'd     RW    4315
0146 Not Visible             01C:0 16B:C6  BCV           N/Asst'd     RW    4315
0147 Not Visible             01C:0 02B:C6  BCV           N/Asst'd     RW    4315
0148 Not Visible             01C:0 01A:C2  BCV           N/Asst'd     RW    4315
0149 Not Visible             01C:0 15A:C2  BCV           N/Asst'd     RW    4315
014A Not Visible             01C:0 16A:C1  BCV           N/Asst'd     RW    4315
014B Not Visible             01C:0 02A:C1  BCV           N/Asst'd     RW    4315
014C Not Visible             01C:0 01B:CD  BCV           N/Asst'd     RW    4315
014D Not Visible             01C:0 15B:CD  BCV           N/Asst'd     RW    4315
014E Not Visible             01C:0 16B:CA  BCV           N/Asst'd     RW    4315
014F Not Visible             01C:0 02B:CA  BCV           N/Asst'd     RW    4315
```

- The `symbcv` command with the `-rdf` and `-bcv` options associates the *remote* BCV devices with the device group.

  `symbcv -g symrep associateall dev -range 137:14F -bcv -rdf`

- The `symdg show` command displays detailed group information about device group symrep. The group contains 22 local standard devices, 22 local R1 BCVs, and 22 remote BCVs.

symdg show symrep

```
Group Name:  symrep

    Group Type                                   : REGULAR
    Device Group in GNS                          : No
    Valid                                        : Yes
    Symmetrix ID                                 : 000187900035
    Group Creation Time                          : Mon Nov 17 11:27:08 2003
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :   22
    Number of Associated GK's                    :    0
    Number of Locally-associated BCV's           :   22
    Number of Locally-associated VDEV's          :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):   22
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (22):
        {
        --------------------------------------------------------------------
                                                    Sym                  Cap
        LdevName              PdevName              Dev  Att. Sts       (MB)
        --------------------------------------------------------------------
        DEV001                /dev/vx/rdmp/c15t2d2s2  0056 (M)  RW     12946
        DEV002                /dev/vx/rdmp/c15t2d3s2  0059 (M)  RW      8631
        DEV003                /dev/vx/rdmp/c15t2d4s2  005B      RW      4315
        DEV004                /dev/vx/rdmp/c15t2d5s2  005C      RW      4315
        DEV005                /dev/vx/rdmp/c15t2d6s2  005D      RW      4315
        DEV006                /dev/vx/rdmp/c15t2d7s2  005E      RW      4315
        DEV007                /dev/vx/rdmp/c15t2d8s2  005F      RW      4315
        DEV008                /dev/vx/rdmp/c15t2d9s2  0060      RW      4315
        DEV009                /dev/vx/rdmp/c15t2d10s2 0061      RW      4315
        DEV010                /dev/vx/rdmp/c15t2d11s2 0062      RW      4315
        DEV011                /dev/vx/rdmp/c15t2d12s2 0063      RW      4315
        DEV012                /dev/vx/rdmp/c15t2d13s2 0064      RW      4315
        DEV013                /dev/vx/rdmp/c15t2d14s2 0065      RW      4315
        DEV014                /dev/vx/rdmp/c15t2d15s2 0066      RW      4315
        DEV015                /dev/vx/rdmp/c15t2d16s2 0067      RW      4315
        DEV016                /dev/vx/rdmp/c15t2d17s2 0068      RW      4315
        DEV017                /dev/vx/rdmp/c15t2d18s2 0069      RW      4315
        DEV018                /dev/vx/rdmp/c15t2d19s2 006A      RW      4315
        DEV019                /dev/vx/rdmp/c15t2d20s2 006B      RW      4315
        DEV020                /dev/vx/rdmp/c15t2d21s2 006C      RW      4315
        DEV021                /dev/vx/rdmp/c15t2d22s2 006D      RW      4315
        DEV022                /dev/vx/rdmp/c15t2d23s2 006E      RW      4315
        }

    BCV Devices Locally-associated (22):
        {
        --------------------------------------------------------------------
                                                    Sym                  Cap
        LdevName              PdevName              Dev  Att. Sts       (MB)
        --------------------------------------------------------------------
        BCV001                N/A                   0182 (M)  NR     12946
        BCV002                N/A                   0185 (M)  NR      8631
        BCV003                N/A                   0187      NR      4315
        BCV004                N/A                   0188      NR      4315
        BCV005                N/A                   0189      NR      4315
        BCV006                N/A                   018A      NR      4315
```

```
        BCV007                N/A                       018B      NR       4315
        BCV008                N/A                       018C      NR       4315
        BCV009                N/A                       018D      NR       4315
        BCV010                N/A                       018E      NR       4315
        BCV011                N/A                       018F      NR       4315
        BCV012                N/A                       0190      NR       4315
        BCV013                N/A                       0191      NR       4315
        BCV014                N/A                       0192      NR       4315
        BCV015                N/A                       0193      NR       4315
        BCV016                N/A                       0194      NR       4315
        BCV017                N/A                       0195      NR       4315
        BCV018                N/A                       0196      NR       4315
        BCV019                N/A                       0197      NR       4315
        BCV020                N/A                       0198      NR       4315
        BCV021                N/A                       0199      NR       4315
        BCV022                N/A                       019A      NR       4315
        }

    BCV Devices Remotely-associated (BCV RDF) (22):
        {
        --------------------------------------------------------------------
                                                        Sym                 Cap
        LdevName              PdevName                  Dev  Att. Sts      (MB)
        --------------------------------------------------------------------
        BRBCV001              N/A                       0137 (M)  RW      12946
        BRBCV002              N/A                       013A (M)  RW       8631
        BRBCV003              N/A                       013C      RW       4315
        BRBCV004              N/A                       013D      RW       4315
        BRBCV005              N/A                       013E      RW       4315
        BRBCV006              N/A                       013F      RW       4315
        BRBCV007              N/A                       0140      RW       4315
        BRBCV008              N/A                       0141      RW       4315
        BRBCV009              N/A                       0142      RW       4315
        BRBCV010              N/A                       0143      RW       4315
        BRBCV011              N/A                       0144      RW       4315
        BRBCV012              N/A                       0145      RW       4315
        BRBCV013              N/A                       0146      RW       4315
        BRBCV014              N/A                       0147      RW       4315
        BRBCV015              N/A                       0148      RW       4315
        BRBCV017              N/A                       014A      RW       4315
        BRBCV018              N/A                       014B      RW       4315
        BRBCV019              N/A                       014C      RW       4315
        BRBCV020              N/A                       014D      RW       4315
        BRBCV021              N/A                       014E      RW       4315
        BRBCV022              N/A                       014F      RW       4315
        }

    Device Group BCV RDF Information
        {
        RDF Type                                : R1
        RDF (RA) Group Number                   : 1                (00)

        Remote Symmetrix ID                     : 000187900041

        R2 Device Is Larger Than The R1 Device : False

        RDF Mode                                : Synchronous
        RDF Adaptive Copy                       : Disabled
        RDF Adaptive Copy Write Pending State   : N/A
        RDF Adaptive Copy Skew (Tracks)         : 65535

        RDF Device Domino                       : Disabled

        RDF Link Configuration                  : Fibre
        RDF Link Domino                         : Disabled
        Prevent Automatic RDF Link Recovery     : Disabled
```

```
Prevent RAs Online Upon Power ON       : Enabled

Device RDF Status                      : Ready          (RW)

Device RA Status                       : Ready          (RW)
Device Link Status                     : Not Ready      (NR)

Device Suspend State                   : Offline
Device Consistency State               : Disabled
RDF R2 Not Ready If Invalid            : Enabled

Device RDF State                       : Not Ready      (NR)
Remote Device RDF State                : Not Ready      (NR)

RDF Pair State ( R1 <-  -> R2 )        : Suspended

Number of R1 Invalid Tracks            : 0
Number of R2 Invalid Tracks            : 0
}
```

- The following command illustrates the use of the vi text editor to create a text file named `symrep.opt`. As was done here, you can enter into the file those parameters and values that specify the single hop configuration and define copy cycle parameters for use during the symreplicate session: one cycle with a duration of 10 minutes. The `CYCLE_OVERFLOW` value of `NEXT` has no relevance here in a setup that has only one copy cycle, but this value will play a role later when the file is edited to have two copy cycles.

  `vi symrep.opt`

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=1
```

- The `symreplicate setup` command performs the setup required to begin a replicate session. The difference between this command or using the `-setup` option with the `symreplicate start` command is that the latter will cycle as many times as you have specified in your symreplicate options file, whereas the `setup` command will cycle just once. If you have various "wait" options in the options file, the setup honors them.

  `symreplicate -g symrep setup -optimize -options symrep.opt -foreground -nop`

```
Checking for valid group configuration...

Checking for valid initial group state...

Setting up local BCV pairs...

Optimizing Local BCV pairs...

Waiting for local BCV synchronization...

Splitting local BCV pairs...

Incrementally establishing RDF pairs...

Setting up remote BCV pairs...

Optimizing remote BCV pairs...

Incrementally establishing remote BCV pairs...

Waiting for remote device synchronization...

Waiting for RDF synchronization...
```

```
Splitting remote BCV pairs...

Incrementally establishing local BCV pairs...

Setup complete; exiting symreplicate...
```

- The following command runs a symreplicate session in the foreground so that the resulting output display illustrates the various steps involved in completing one copy cycle. Note that the symrep.opt file specified on the command line tells SYMCLI what copy cycle parameters to employ during the session. The `-consistent` option performs a consistent split of the local BCV pairs during the cycle.

symreplicate -g symrep start -options symrep.opt -foreground -consistent -nop

```
Checking for valid group configuration...

Checking for valid initial group state...

Waiting for local BCV synchronization...

Splitting local BCV pairs...

Incrementally establishing RDF pairs...

Waiting for RDF synchronization...

Suspending RDF connection...

Waiting for local BCV pairs to split...

Incrementally establishing local BCV pairs...

Incrementally establishing remote BCV pairs...

Waiting for remote device synchronization...

Splitting remote BCV pairs...

1 cycle(s) complete; exiting symreplicate...
```

- This `symreplicate` command runs one copy cycle in the background (the default when the `-foreground` option is omitted). A subsequent `symreplicate query` will check the status of the cycle being processed in the background.

  symreplicate -g symrep start -options symrep.opt -noprompt

```
Checking for valid group configuration...

Checking for valid initial group state...

symreplicate process launched.
```

- The `symreplicate query` command checks the status of the copy cycle being processed in the background. SYMCLI provides an updated display every five seconds. Only a representative sample of the update displays is shown below.

`symreplicate -g symrep query -i 5`

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Waiting for next cycle              0 m        1        1

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing RDF pairs              0 m        1        1

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing local and remote       0 m        1        1

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Completed Complete                            0 m        1        1
```

- The following command uses the vi text editor again to edit the text file named `symrep.opt`. As was done here, you can edit parameter values that affect the symreplicate session. By changing the number of copy cycles from one to two, the `CYCLE_OVERFLOW` value of `NEXT` becomes relevant. If the first copy cycle lasts longer than its 10-minute time schedule, the second copy cycle will begin at the next scheduled start. For example, if the first copy cycle overflows to 15 minutes, the second cycle begins at the 20-minute mark. If the first copy cycle overflows to 35 minutes, the second cycle begins at the 40-minute mark.

  `vi symrep.opt`

  ```
  SYMCLI_REPLICATE_HOP_TYPE=SINGLE
  SYMCLI_REPLICATE_CYCLE=10
  SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
  SYMCLI_REPLICATE_NUM_CYCLES=2
  ```

- This `symreplicate start` command runs the two-cycle session in the background.

  `symreplicate -g symrep start -options symrep.opt -noprompt`

  ```
  Checking for valid group configuration...

  Checking for valid initial group state...
  ```

- The `symreplicate query` command checks the status of the copy cycles being processed in the background.

```
symreplicate -g symrep query

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                                Cycle    Current  Max
Hop Type  Status    Step                                 Period   Cycle    Cycles
--------- --------- ------------------------------ -------- -------- --------
SINGLE    Active    Establishing RDF pairs               10 m        1        2
```

- The `symreplicate stop` command stops the current replicate session. The `-step` option causes the stop to occur after the current execution step completes. Omitting `-step` would stop the session at the end of a complete copy cycle.

```
symreplicate -g symrep stop -step -noprompt

Stop operation underway.
```

- The `symreplicate query` command checks the status of the session. The display indicates that the session stopped at the step for establishing the RDF pairs.

```
symreplicate -g symrep query

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                                Cycle    Current  Max
Hop Type  Status    Step                                 Period   Cycle    Cycles
--------- --------- ------------------------------ -------- -------- --------
SINGLE    Stopped   Establishing RDF pairs               10 m        1        2
```

- The `symreplicate restart` command resumes the copy cycle at the step where the session stopped. Specifying the options file again on `restart` is not required unless you changed the file while the session was stopped. Although the example made no changes to the options file here, the options file is specified again for consistency.

```
symreplicate -g symrep restart -options symrep.opt -noprompt

symreplicate process launched.
```

- The `symreplicate query` command checks the status of the copy cycle every five seconds and provides an updated display. Note that the cycle resumes where it was in the sequence of steps when the session stopped. Only a representative sample of the update displays is shown below.

```
symreplicate -g symrep query -i 5

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing RDF pairs              10 m        1        2

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing local and remote      10 m        1        2

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Waiting for next cycle             10 m        1        2

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing RDF pairs             10 m        2        2

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Active    Establishing local and remote      10 m        2        2

Device Group (DG) Name      : symrep
DG's Symmetrix ID           : 000187900035
Remote Symmetrix ID         : 000187900041

Replicate                                               Cycle    Current  Max
Hop Type  Status    Step                                Period   Cycle    Cycles
--------- --------- ----------------------------- -------- -------- --------
SINGLE    Completed Completed                          10 m        2        2
```

Example 2: SRDF/AR Multi-Hop Configuration with BCVs at Hop 2

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host connected to a source Symmetrix (sid 79). All commands are issued from the source-side host. The example uses the following devices to create the multi-hop environment and shows how to set up the correct pair states for automated data replication in this environment:

- Source Symmetrix (sid 79): R1 devices CEE-CFD
- Hop 1 Symmetrix (sid 67): R2 devices EA2-EB1; R1 BCV devices E92-EA1
- Hop 2 Symmetrix (sid 01): R2 devices 318-327; BCV devices 328-337
- The `symrdf list` command shows the local view of the source R1 devices (SymDev), their target R2 mirror devices (RDev), and their current SRDF pair state. The ellipsis (…) represents truncated output.

`symrdf list -r1 -sid 79`

```
Symmetrix ID: 000187700079

                              Local Device View
-------------------------------------------------------------------------
                   STATUS    MODES                    RDF  S T A T E S
Sym        RDF    --------- -----  R1 Inv    R2 Inv ----------------------
Dev  RDev  Typ:G  SA RA LNK  MDA   Tracks    Tracks Dev RDev Pair
---- ---- ------ --------- ----- -------   ------- --- ---- -------------
0CEE 0EA2  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CEF 0EA3  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF0 0EA4  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF1 0EA5  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF2 0EA6  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF3 0EA7  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF4 0EA8  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF5 0EA9  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF6 0EAA  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF7 0EAB  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF8 0EAC  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CF9 0EAD  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CFA 0EAE  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CFB 0EAF  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CFC 0EB0  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
0CFD 0EB1  R1:64 RW RW RW   S..         0         0 RW  WD   Synchronized
.........................................................................
   …
```

- The `symdg create` command creates a device group (symrep). The `symld addall` command adds the RDF standard devices from local Symmetrix 000187700079 to the group, using the `-range` option to limit the selections to those devices between CEE and CFD. The symbcv command with the `-rdf` option associates a range of remote BCV devices on Hop 1 with the device group. The symbcv command with the `-rrdf` option associates a range of remote BCV devices on Hop 2 with the device group.

```
symdg create symrep -type rdf1
symld -g symrep addall dev -range CEE:CFD -sid 79
symbcv -g symrep -rdf associateall dev -range E92:EA1
symbcv -g symrep -rrdf associateall dev -range 328:337
```

- The following command illustrates the use of the vi text editor to create a text file named `rep_opt.txt`. As was done here, you can enter into the file those parameters and values that specify the multi-hop configuration that uses the Hop 2 BCVs (USE_FINAL_BCV=TRUE) and define copy cycle parameters for use during the symreplicate session. The CYCLE_OVERFLOW value of NEXT has no relevance here in a setup that has only one copy cycle, but this value can play a role later if the file is edited to have more than one copy cycle.

  `vi rep_opt.txt`

```
SYMCLI_REPLICATE_HOP_TYPE=MULTI
SYMCLI_REPLICATE_USE_FINAL_BCV=TRUE
SYMCLI_REPLICATE_CYCLE=0
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=1
```

- The `symreplicate start` command with the `-setup` option sets up the required pair states, and if successful, begins the symreplicate session. This command will cycle as many times as you have specified in your symreplicate options file (the `symreplicate setup` command cycles just once).

  `symreplicate -g symrep start -setup -optimize -options rep_opt.txt -foreground -noprompt`

```
Checking for valid group configuration...

Checking for valid initial group state...

Setting up local RDF pairs...

Setting up first hop BCV pairs...

Optimizing first hop BCV pairs...

Waiting for first hop BCV device synchronization...

Splitting first hop BCV pairs...

Incrementally establishing remote RDF pairs...

Setting up second hop BCV pairs...

Optimizing second hop BCV pairs...

Incrementally establishing second hop BCV pairs...

Waiting for second hop BCV synchronization...

Waiting for remote RDF pair synchronization...

Splitting second hop BCV pairs...

Incrementally establishing first hop BCV pairs...

Setup is complete...

1 cycle(s) complete; exiting symreplicate...
```

Example 3: Setting Up an SRDF/AR Single-Hop Configuration Using a CG

This example is performed using Solutions Enabler version 5.4. The hardware setup illustrates a single-hop configuration in which the source devices span three Symmetrix arrays (SIDs 35, 43, and 60). A composite group is defined on a host connected to these three Symmetrix arrays. The devices include standard devices and R1 BCV devices from the local Symmetrix arrays, as well as BCVs from the remote Symmetrix arrays.

- The `symcg create` command creates a Regular type composite group named single-hop.

  ```
  symcg create single-hop -type regular
  ```

- The following `symcg` commands add to the composite group a range of standard devices from each of the three local source Symmetrix arrays.

  ```
  symcg -cg single-hop addall dev -range 56:6E -sid 35
  symcg -cg single-hop addall dev -range 61:79 -sid 43
  symcg -cg single-hop addall dev -range 14:27 -sid 60
  ```

- The following `symbcv` commands associate with the composite group a range of R1 BCV devices from each of the three local source Symmetrix arrays.

  ```
  symbcv -cg single-hop associateall dev -range 182:19A -sid 35
  symbcv -cg single-hop associateall dev -range 142:15A -sid 43
  symbcv -cg single-hop associateall dev -range 3B6:3C9 -sid 60
  ```

- The following `symbcv` commands with the `-rdf` option associate a range of BCV devices from each of the *remote* Symmetrix arrays. The `-bcv` option specifies that the source devices are local R1 BCV devices. If there is more than one RDF group on a local Symmetrix array, you must include the RDF group number of the local source devices (the group number of the R1 BCVs). Specifying the RDF group number creates each R1/R2 pairing as well as the RDF link for that pair.

  ```
  symbcv -cg single-hop associateall dev -range 137:14F -bcv -rdf
     -sid 35 -rdfg 1
  symbcv -cg single-hop associateall dev -range 12A:142 -bcv -rdf
     -sid 43 -rdfg 1
  symbcv -cg single-hop associateall dev -range 21C:22F -bcv -rdf
     -sid 60 -rdfg 1
  ```

- The `symcg list` command displays a list of composite groups defined on this host.

```
symcg list

                     C O M P O S I T E   G R O U P S

                                             Number of           Number of
    Name                    Type     Valid  Symms    RAGs    Devs   BCVs   VDEVs

    single-hop              REGULAR Yes         3       3      64    128        0
```

- The `symcg show` command displays detailed configuration and status information about the composite group.

```
symcg show single-hop

Composite Group Name:  single-hop

  Composite Group Type                                 : REGULAR
  Valid                                                : Yes
  CG in PowerPath                                      : No
  CG in GNS                                            : No

  Number of RDF (RA) Groups                            :     3
  Number of STD Devices                                :    64
```

```
Number of BCV's (Locally-associated)              :   64
Number of VDEV's (Locally-associated)             :    0
Number of RBCV's (Remotely-associated STD-RDF)    :    0
Number of BRBCV's (Remotely-associated BCV-RDF)   :   64
Number of RRBCV's (Remotely-associated RBCV)      :    0

Number of Symmetrix Units (3):
   {

   1) Symmetrix ID                                      : 000187900035
      Microcode Version                                 : 5670
      Number of STD Devices                             :   22
      Number of BCV's (Locally-associated)              :   22
      Number of VDEV's (Locally-associated)             :    0
      Number of RBCV's (Remotely-associated STD_RDF)    :    0
      Number of BRBCV's (Remotely-associated BCV-RDF):   22
      Number of RRBCV's (Remotely-associated RBCV)      :    0

      Number of RDF (RA) Groups (1):
         {

         1) RDF (RA) Group Number  :  1            (00)
            Remote Symmetrix ID    : 000187900041
            Microcode Version      : 5670

            BCV's (Locally-associated) (22):
               {
              -------------------------------------------------------------
                                       Sym  Device        Consistency   Cap
               PdevName                Dev  Config        State         (MB)
              -------------------------------------------------------------
               N/A                     0182 RDF1-BCV+Mir  Disabled     12946
               N/A                     0185 RDF1-BCV+Mir  Disabled      8631
               N/A                     0187 RDF1-BCV+Mir  Disabled      4315
               N/A                     0188 RDF1-BCV+Mir  Disabled      4315
               N/A                     0189 RDF1-BCV+Mir  Disabled      4315
               N/A                     018A RDF1-BCV+Mir  Disabled      4315
               N/A                     018B RDF1-BCV+Mir  Disabled      4315
               N/A                     018C RDF1-BCV+Mir  Disabled      4315
               N/A                     018D RDF1-BCV+Mir  Disabled      4315
               N/A                     018E RDF1-BCV+Mir  Disabled      4315
               N/A                     018F RDF1-BCV+Mir  Disabled      4315
               N/A                     0190 RDF1-BCV+Mir  Disabled      4315
               N/A                     0191 RDF1-BCV+Mir  Disabled      4315
               N/A                     0192 RDF1-BCV+Mir  Disabled      4315
               N/A                     0193 RDF1-BCV+Mir  Disabled      4315
               N/A                     0194 RDF1-BCV+Mir  Disabled      4315
               N/A                     0195 RDF1-BCV+Mir  Disabled      4315
               N/A                     0196 RDF1-BCV+Mir  Disabled      4315
               N/A                     0197 RDF1-BCV+Mir  Disabled      4315
               N/A                     0198 RDF1-BCV+Mir  Disabled      4315
               N/A                     0199 RDF1-BCV+Mir  Disabled      4315
               N/A                     019A RDF1-BCV+Mir  Disabled      4315
               }

            BRBCV's (Remotely-associated BCV-RDF) (22):
               {
              -------------------------------------------------------------
                                       Sym  Device        Consistency   Cap
               PdevName                Dev  Config        State         (MB)
              -------------------------------------------------------------
               N/A                     0137 BCV           N/A          12946
               N/A                     013A BCV           N/A           8631
               N/A                     013C BCV           N/A           4315
               N/A                     013D BCV           N/A           4315
               N/A                     013E BCV           N/A           4315
```

```
                N/A                     013F BCV          N/A           4315
                N/A                     0140 BCV          N/A           4315
                N/A                     0141 BCV          N/A           4315
                N/A                     0142 BCV          N/A           4315
                N/A                     0143 BCV          N/A           4315
                N/A                     0144 BCV          N/A           4315
                N/A                     0145 BCV          N/A           4315
                N/A                     0146 BCV          N/A           4315
                N/A                     0147 BCV          N/A           4315
                N/A                     0148 BCV          N/A           4315
                N/A                     0149 BCV          N/A           4315
                N/A                     014A BCV          N/A           4315
                N/A                     014B BCV          N/A           4315
                N/A                     014C BCV          N/A           4315
                N/A                     014D BCV          N/A           4315
                N/A                     014E BCV          N/A           4315
                N/A                     014F BCV          N/A           4315
                 }
           }

        STD Devices (non-RDF) (22):
           {
           --------------------------------------------------------------
                                   Sym  Device        Consistency    Cap
           PdevName                Dev  Config        State          (MB)
           --------------------------------------------------------------
           /dev/vx/rdmp/c15t2d2s2  0056 2-Way Mir     N/A          12946
           /dev/vx/rdmp/c15t2d3s2  0059 2-Way Mir     N/A           8631
           /dev/vx/rdmp/c15t2d4s2  005B 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d5s2  005C 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d6s2  005D 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d7s2  005E 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d8s2  005F 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d9s2  0060 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d10s2 0061 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d11s2 0062 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d12s2 0063 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d13s2 0064 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d14s2 0065 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d15s2 0066 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d16s2 0067 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d17s2 0068 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d18s2 0069 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d19s2 006A 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d20s2 006B 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d21s2 006C 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d22s2 006D 2-Way Mir     N/A           4315
           /dev/vx/rdmp/c15t2d23s2 006E 2-Way Mir     N/A           4315
           }

     2) Symmetrix ID                                    : 000000003143
        Microcode Version                               : 5267
        Number of STD Devices                           :   22
        Number of BCV's (Locally-associated)            :   22
        Number of VDEV's (Locally-associated)           :    0
        Number of RBCV's (Remotely-associated STD_RDF)  :    0
        Number of BRBCV's (Remotely-associated BCV-RDF) :   22
        Number of RRBCV's (Remotely-associated RBCV)    :    0

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number  :  1            (A)
              Remote Symmetrix ID    : 000000003156
              Microcode Version      : 5267
```

```
            BCV's (Locally-associated) (22):
              {
              ------------------------------------------------------------
                                      Sym  Device        Consistency    Cap
               PdevName               Dev  Config        State         (MB)
              ------------------------------------------------------------
               N/A                    0142 RDF1-BCV      Disabled     12946
               N/A                    0145 RDF1-BCV      Disabled      8631
               N/A                    0147 RDF1-BCV      Disabled      4315
               N/A                    0148 RDF1-BCV      Disabled      4315
               N/A                    0149 RDF1-BCV      Disabled      4315
               N/A                    014A RDF1-BCV      Disabled      4315
               N/A                    014B RDF1-BCV      Disabled      4315
               N/A                    014C RDF1-BCV      Disabled      4315
               N/A                    014D RDF1-BCV      Disabled      4315
               N/A                    014E RDF1-BCV      Disabled      4315
               N/A                    014F RDF1-BCV      Disabled      4315
               N/A                    0150 RDF1-BCV      Disabled      4315
               N/A                    0151 RDF1-BCV      Disabled      4315
               N/A                    0152 RDF1-BCV      Disabled      4315
               N/A                    0153 RDF1-BCV      Disabled      4315
               N/A                    0154 RDF1-BCV      Disabled      4315
               N/A                    0155 RDF1-BCV      Disabled      4315
               N/A                    0156 RDF1-BCV      Disabled      4315
               N/A                    0157 RDF1-BCV      Disabled      4315
               N/A                    0158 RDF1-BCV      Disabled      4315
               N/A                    0159 RDF1-BCV      Disabled      4315
               N/A                    015A RDF1-BCV      Disabled      4315
              }

            BRBCV's (Remotely-associated BCV-RDF) (22):
              {
              ------------------------------------------------------------
                                      Sym  Device        Consistency    Cap
               PdevName               Dev  Config        State         (MB)
              ------------------------------------------------------------
               N/A                    012A BCV           N/A          12946
               N/A                    012D BCV           N/A           8631
               N/A                    012F BCV           N/A           4315
               N/A                    0130 BCV           N/A           4315
               N/A                    0131 BCV           N/A           4315
               N/A                    0132 BCV           N/A           4315
               N/A                    0133 BCV           N/A           4315
               N/A                    0134 BCV           N/A           4315
               N/A                    0135 BCV           N/A           4315
               N/A                    0136 BCV           N/A           4315
               N/A                    0137 BCV           N/A           4315
               N/A                    0138 BCV           N/A           4315
               N/A                    0139 BCV           N/A           4315
               N/A                    013A BCV           N/A           4315
               N/A                    013B BCV           N/A           4315
               N/A                    013C BCV           N/A           4315
               N/A                    013D BCV           N/A           4315
               N/A                    013E BCV           N/A           4315
               N/A                    013F BCV           N/A           4315
               N/A                    0140 BCV           N/A           4315
               N/A                    0141 BCV           N/A           4315
               N/A                    0142 BCV           N/A           4315
              }
         }

      STD Devices (non-RDF) (22):
         {
         ------------------------------------------------------------
                                 Sym  Device        Consistency    Cap
         PdevName                Dev  Config        State         (MB)
```

```
        -------------------------------------------------------------
        /dev/vx/rdmp/c15t1d2s2  0061 2-Way Mir     N/A          12946
        /dev/vx/rdmp/c15t1d3s2  0064 2-Way Mir     N/A           8631
        /dev/vx/rdmp/c15t1d4s2  0066 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d5s2  0067 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d6s2  0068 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d7s2  0069 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d8s2  006A 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d9s2  006B 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d10s2 006C 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d11s2 006D 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d12s2 006E 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d13s2 006F 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d14s2 0070 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d15s2 0071 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d16s2 0072 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d17s2 0073 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d18s2 0074 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d19s2 0075 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d20s2 0076 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d21s2 0077 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d22s2 0078 2-Way Mir     N/A           4315
        /dev/vx/rdmp/c15t1d23s2 0079 2-Way Mir     N/A           4315
        }

  3) Symmetrix ID                                    : 000184500160
     Microcode Version                               : 5568
     Number of STD Devices                           :   20
     Number of BCV's (Locally-associated)            :   20
     Number of VDEV's (Locally-associated)           :    0
     Number of RBCV's (Remotely-associated STD_RDF)  :    0
     Number of BRBCV's (Remotely-associated BCV-RDF):   20
     Number of RRBCV's (Remotely-associated RBCV)    :    0

     Number of RDF (RA) Groups (1):
        {

        1) RDF (RA) Group Number  :  1             (A)
           Remote Symmetrix ID    : 000184502898
           Microcode Version      : 5568

           BCV's (Locally-associated) (20):
              {
            -------------------------------------------------------------
                                    Sym  Device        Consistency   Cap
             PdevName               Dev  Config        State        (MB)
            -------------------------------------------------------------
             N/A                    03B6 RDF1-BCV+Mir  Disabled      4315
             N/A                    03B7 RDF1-BCV+Mir  Disabled      4315
             N/A                    03B8 RDF1-BCV+Mir  Disabled      4315
             N/A                    03B9 RDF1-BCV+Mir  Disabled      4315
             N/A                    03BA RDF1-BCV+Mir  Disabled      4315
             N/A                    03BB RDF1-BCV+Mir  Disabled      4315
             N/A                    03BC RDF1-BCV+Mir  Disabled      4315
             N/A                    03BD RDF1-BCV+Mir  Disabled      4315
             N/A                    03BE RDF1-BCV+Mir  Disabled      4315
             N/A                    03BF RDF1-BCV+Mir  Disabled      4315
             N/A                    03C0 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C1 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C2 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C3 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C4 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C5 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C6 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C7 RDF1-BCV+Mir  Disabled      4315
             N/A                    03C8 RDF1-BCV+Mir  Disabled      4315
```

```
                    N/A                           03C9 RDF1-BCV+Mir  Disabled      4315
                     }

                  BRBCV's (Remotely-associated BCV-RDF) (20):
                     {
                   -------------------------------------------------------------
                                                  Sym  Device        Consistency   Cap
                    PdevName                      Dev  Config        State         (MB)
                   -------------------------------------------------------------
                    N/A                           021C 2-Way BCV Mir N/A           4315
                    N/A                           021D 2-Way BCV Mir N/A           4315
                    N/A                           021E 2-Way BCV Mir N/A           4315
                    N/A                           021F 2-Way BCV Mir N/A           4315
                    N/A                           0220 2-Way BCV Mir N/A           4315
                    N/A                           0221 2-Way BCV Mir N/A           4315
                    N/A                           0222 2-Way BCV Mir N/A           4315
                    N/A                           0223 2-Way BCV Mir N/A           4315
                    N/A                           0224 2-Way BCV Mir N/A           4315
                    N/A                           0225 2-Way BCV Mir N/A           4315
                    N/A                           0226 2-Way BCV Mir N/A           4315
                    N/A                           0227 2-Way BCV Mir N/A           4315
                    N/A                           0228 2-Way BCV Mir N/A           4315
                    N/A                           0229 2-Way BCV Mir N/A           4315
                    N/A                           022A 2-Way BCV Mir N/A           4315
                    N/A                           022B 2-Way BCV Mir N/A           4315
                    N/A                           022C 2-Way BCV Mir N/A           4315
                    N/A                           022D 2-Way BCV Mir N/A           4315
                    N/A                           022E 2-Way BCV Mir N/A           4315
                    N/A                           022F 2-Way BCV Mir N/A           4315
                     }
                }

          STD Devices (non-RDF) (20):
                {
                -------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                PdevName                Dev  Config        State         (MB)
                -------------------------------------------------------------
                /dev/rdsk/emcpower38c   0014 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower39c   0015 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower40c   0016 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower41c   0017 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower42c   0018 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower43c   0019 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower44c   001A 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower45c   001B 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower46c   001C 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower47c   001D 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower48c   001E 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower49c   001F 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower50c   0020 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower51c   0021 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower52c   0022 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower53c   0023 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower54c   0024 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower55c   0025 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower56c   0026 2-Way Mir     N/A           4315
                /dev/rdsk/emcpower57c   0027 2-Way Mir     N/A           4315
                }
     }
```

- The following command illustrates the use of the vi text editor to create a text file named `sar.opt`. The setup operation requires that the `HOP_TYPE` be defined. You define copy cycle parameters for use during the symreplicate session (for example, two cycles where the time from the beginning of the first cycle to the beginning of the next cycle should be 10 minutes).

  `vi sar.opt`

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=2
```

- Similar to Example 1, the `symreplicate setup` command performs the setup here for all pairs in the composite group that spans three source Symmetrix arrays. This command results in the setup performing one cycle.

  `symreplicate -cg single-hop setup -foreground -options sar.opt -noprompt`

```
Checking for valid group configuration...

Checking for valid initial group state...

Setting up local BCV pairs...

Waiting for local BCV synchronization...

Splitting local BCV pairs...

Incrementally establishing RDF pairs...

Setting up remote BCV pairs...

Incrementally establishing remote BCV pairs...

Waiting for remote device synchronization...

Waiting for RDF synchronization...

Splitting remote BCV pairs...

Incrementally establishing local BCV pairs...

Setup complete; exiting symreplicate...
```

- The setup is complete. You can now perform `symreplicate start` for the composite group as was done for the device group in Example 1.

Example 4: Setting Up an SRDF/AR Multi-Hop Configuration Using a CG

This example is performed using Solutions Enabler version 5.4. The hardware setup illustrates a multi-hop configuration in which the source devices span two Symmetrix arrays (SIDs 79 and 32). A composite group is defined on a Solaris host connected to these two Symmetrix arrays. The devices include R1 devices from the local Symmetrix arrays, as well as BCVs from on the remote Hop-1 and Hop-2 Symmetrix arrays.

- The `symcg create` command creates an RDF1 type composite group named multi-hop.

  ```
  symcg create multi-hop -type rdf1
  ```

- The following `symcg` commands add to the composite group a range of standard devices from each of the two local source Symmetrix arrays.

  ```
  symcg -cg multi-hop addall dev -range CEE:CFD -sid 79
  symcg -cg multi-hop addall dev -range 2:7 -sid 32
  ```

- The following `symbcv` commands with the `-rdf` option associate a range of BCV devices from each of the *remote* Hop-1 Symmetrix arrays. If there is more than one RDF group on a local Symmetrix array, you must include the RDF group number of the local R1 source devices. Specifying the RDF group number creates the RDF link for each R1/R2 pair.

  ```
  symbcv -cg multi-hop associateall dev -range E92:EA1 -rdf -rdfg 64 -sid 79
  symbcv -cg multi-hop associateall dev -range 48:4D -rdf -rdfg 1 -sid 32
  ```

- The following `symbcv` commands with the `-rrdf` option associate a range of BCV devices from each of the *remote* Hop-2 Symmetrix arrays. To define the path of the RDF link, you must always include the group number of the local source devices as specified with the `-rdfg` option.[1]

  ```
  symbcv -cg multi-hop associateall dev -range 328:337 -rrdf -rdfg 64 -sid 79
  symbcv -cg multi-hop associateall dev -range 30:35 -rrdf -rdfg 1 -sid 32
  ```

- The `symcg list` command displays a list of composite groups defined on this host.

  ```
  symcg list
  ```

```
                        C O M P O S I T E   G R O U P S

                                              Number of           Number of
Name                     Type     Valid   Symms    RAGs     Devs    BCVs    VDEVs

multi-hop                RDF1     Yes         2       2       22      44        0
```

1. If you add the Hop 2 BCVs to the composite group before adding the Hop 1 BCVs, include the `-remote_rdfg` option also.

- The `symcg show` command displays configuration and status information about the composite group.

```
symcg show multi-hop

Composite Group Name:  multi-hop

  Composite Group Type                              : RDF1
  Valid                                             : Yes
  CG in PowerPath                                   : No
  CG in GNS                                         : No

  Number of RDF (RA) Groups                         :    2
  Number of STD Devices                             :   22
  Number of BCV's (Locally-associated)              :    0
  Number of VDEV's (Locally-associated)             :    0
  Number of RBCV's (Remotely-associated STD-RDF)    :   22
  Number of BRBCV's (Remotely-associated BCV-RDF)   :    0
  Number of RRBCV's (Remotely-associated RBCV)      :   22

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                      : 000187700079
        Microcode Version                                 : 5670
        Number of STD Devices                             :   16
        Number of BCV's (Locally-associated)              :    0
        Number of VDEV's (Locally-associated)             :    0
        Number of RBCV's (Remotely-associated STD_RDF)  :   16
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)    :   16

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number   : 64              (3F)
              Remote Symmetrix ID     : 000187700067
              Microcode Version       : 5670

              STD Devices (16):
                 {
               -------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                PdevName                Dev  Config        State         (MB)
               -------------------------------------------------------------
                /dev/vx/rdmp/c15t0d63s2 0CEE RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d64s2 0CEF RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d65s2 0CF0 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d66s2 0CF1 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d67s2 0CF2 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d68s2 0CF3 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d69s2 0CF4 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d70s2 0CF5 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d71s2 0CF6 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d72s2 0CF7 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d73s2 0CF8 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d74s2 0CF9 RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d75s2 0CFA RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d76s2 0CFB RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d77s2 0CFC RDF1+Mir      Disabled      4315
                /dev/vx/rdmp/c15t0d78s2 0CFD RDF1+Mir      Disabled      4315
                 }

              RBCV's (Remotely-associated STD-RDF) (16):
                 {
                 Remote RDF (RA) Group Number   : 52            (33)
```

```
                Remote Remote Symmetrix ID    : 000000006201
                Microcode Version             : 5670

               -------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                PdevName                Dev  Config        State         (MB)
               -------------------------------------------------------------
                N/A                     0E92 RDF1-BCV      Disabled      4315
                N/A                     0E93 RDF1-BCV      Disabled      4315
                N/A                     0E94 RDF1-BCV      Disabled      4315
                N/A                     0E95 RDF1-BCV      Disabled      4315
                N/A                     0E96 RDF1-BCV      Disabled      4315
                N/A                     0E97 RDF1-BCV      Disabled      4315
                N/A                     0E98 RDF1-BCV      Disabled      4315
                N/A                     0E99 RDF1-BCV      Disabled      4315
                N/A                     0E9A RDF1-BCV      Disabled      4315
                N/A                     0E9B RDF1-BCV      Disabled      4315
                N/A                     0E9C RDF1-BCV      Disabled      4315
                N/A                     0E9D RDF1-BCV      Disabled      4315
                N/A                     0E9E RDF1-BCV      Disabled      4315
                N/A                     0E9F RDF1-BCV      Disabled      4315
                N/A                     0EA0 RDF1-BCV      Disabled      4315
                N/A                     0EA1 RDF1-BCV      Disabled      4315
                }

             RRBCV's (Remotely-associated RBCV) (16):
                {
               -------------------------------------------------------------
                                        Sym  Device        Consistency   Cap
                PdevName                Dev  Config        State         (MB)
               -------------------------------------------------------------
                N/A                     0328 BCV           N/A           4315
                N/A                     0329 BCV           N/A           4315
                N/A                     032A BCV           N/A           4315
                N/A                     032B BCV           N/A           4315
                N/A                     032C BCV           N/A           4315
                N/A                     032D BCV           N/A           4315
                N/A                     032E BCV           N/A           4315
                N/A                     032F BCV           N/A           4315
                N/A                     0330 BCV           N/A           4315
                N/A                     0331 BCV           N/A           4315
                N/A                     0332 BCV           N/A           4315
                N/A                     0333 BCV           N/A           4315
                N/A                     0334 BCV           N/A           4315
                N/A                     0335 BCV           N/A           4315
                N/A                     0336 BCV           N/A           4315
                N/A                     0337 BCV           N/A           4315
                }
          }

    2) Symmetrix ID                                      : 000000005232
       Microcode Version                                 : 5568
       Number of STD Devices                             :    6
       Number of BCV's (Locally-associated)              :    0
       Number of VDEV's (Locally-associated)             :    0
       Number of RBCV's (Remotely-associated STD_RDF)    :    6
       Number of BRBCV's (Remotely-associated BCV-RDF)   :    0
       Number of RRBCV's (Remotely-associated RBCV)      :    6

       Number of RDF (RA) Groups (1):
          {

          1) RDF (RA) Group Number  :  1              (A)
             Remote Symmetrix ID    : 000000005231
             Microcode Version      : 5568
```

```
                STD Devices (6):
                   {
                  ----------------------------------------------------------
                                           Sym   Device        Consistency   Cap
                   PdevName                Dev   Config        State         (MB)
                  ----------------------------------------------------------
                   N/A                     0002  RDF1          Disabled      1031
                   N/A                     0003  RDF1          Disabled      1031
                   N/A                     0004  RDF1          Disabled      1031
                   N/A                     0005  RDF1          Disabled      1031
                   N/A                     0006  RDF1          Disabled      1031
                   N/A                     0007  RDF1          Disabled      1031
                   }

                RBCV's (Remotely-associated STD-RDF) (6):
                   {
                   Remote RDF (RA) Group Number   :  2            (B)
                   Remote Remote Symmetrix ID     : 000000005233
                   Microcode Version              : 5568

                  ----------------------------------------------------------
                                           Sym   Device        Consistency   Cap
                   PdevName                Dev   Config        State         (MB)
                  ----------------------------------------------------------
                   N/A                     0048  RDF1-BCV      Disabled      1031
                   N/A                     0049  RDF1-BCV      Disabled      1031
                   N/A                     004A  RDF1-BCV      Disabled      1031
                   N/A                     004B  RDF1-BCV      Disabled      1031
                   N/A                     004C  RDF1-BCV      Disabled      1031
                   N/A                     004D  RDF1-BCV      Disabled      1031
                   }

                RRBCV's (Remotely-associated RBCV) (6):
                   {
                  ----------------------------------------------------------
                                           Sym   Device        Consistency   Cap
                   PdevName                Dev   Config        State         (MB)
                  ----------------------------------------------------------
                   N/A                     0030  BCV           N/A           1031
                   N/A                     0031  BCV           N/A           1031
                   N/A                     0032  BCV           N/A           1031
                   N/A                     0033  BCV           N/A           1031
                   N/A                     0034  BCV           N/A           1031
                   N/A                     0035  BCV           N/A           1031
                   }
             }
        }
```

- The following command illustrates the use of the vi text editor to create a text file named 3-hop.opt. The setup operation requires that the `HOP_TYPE` be defined. By default, `USE_FINAL_BCV` is set to `TRUE`. You define copy cycle parameters for use during the `symreplicate` session (for example, two cycles where the time from the beginning of the first cycle to the beginning of the next cycle should be 10 minutes).

  `vi  3-hop_opt.txt`

  ```
  SYMCLI_REPLICATE_HOP_TYPE=MULTI
  SYMCLI_REPLICATE_USE_FINAL_BCV=TRUE
  SYMCLI_REPLICATE_CYCLE=10
  SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
  SYMCLI_REPLICATE_NUM_CYCLES=2
  ```

- Similar to Example 1, the `symreplicate setup` command performs the setup here for all pairs in the composite group that spans two source Symmetrix arrays. This command results in the setup performing one cycle. The `-optimize` flag is included to optimize BCV device pairings within each Symmetrix array.

`symreplicate -cg multi-hop setup -optimize -foreground -options 3-hop.opt -nop`

```
Checking for valid group configuration...

Checking for valid initial group state...

Setting up local RDF pairs...

Setting up first hop BCV pairs...

Optimizing first hop BCV pairs...

Waiting for first hop BCV device synchronization...

Splitting first hop BCV pairs...

Incrementally establishing remote RDF pairs...

Setting up second hop BCV pairs...

Optimizing second hop BCV pairs...

Incrementally establishing second hop BCV pairs...

Waiting for second hop BCV synchronization...

Waiting for remote RDF pair synchronization...

Splitting second hop BCV pairs...

Incrementally establishing first hop BCV pairs...

Setup complete; exiting symreplicate...
```

- The setup is complete. You can now perform `symreplicate start` for the composite group as was done for the device group in Example 2.

Example 5: Accessing Concurrent Non-SRDF/AR BCVs While Running SRDF/AR (Single-Hop Configuration)[1]

The hardware setup for this single-hop SRDF/AR configuration consists of a host connected to a local source Symmetrix (sid 505) and a remote target Symmetrix. All commands are issued from the source-side host and affect devices on the local Symmetrix. The example establishes a set of non-SRDF/AR BCVs that are concurrent with a set of SRDF/AR BCVs. Devices on the local Symmetrix are:

- SRDF/AR standard devices 012E and 012F
- SRDF/AR BCV devices 04CE and 04CF
- Non-SRDF/AR BCV devices 04CC and 04CD
- The SRDF/AR device group (sar) and the creation of the single-hop environment have already been set up. The SRDF/AR device group was created using the following commands:

```
symdg create sar
symld -g sar addall dev -range 12E:12F
symbcv -g sar associateall dev -range 4CE:4CF
symbcv -g sar associateall dev -range 30C:30D -bcv -rdf
```

- A device file named devfile was created to define the following non-SRDF/AR BCV pairs:

```
012E 04CC
012F 04CD
```

- The following commands were used to fully establish and split the non-SRDF/AR BCV pairs in the device file:

```
symmir -f devfile -sid 505 establish -full -noprompt
symmir -f devfile -sid 505 split -noprompt
```

- The SRDF/AR devices were set up for a SRDF/AR copy cycle using the following commands that operate on the SRDF/AR devices in the device group named sar. (Beginning with Solutions Enabler Version 5.4, you can perform these steps automatically with the `symreplicate setup` command.)

```
symmir -g sar establish -full -exact -noprompt
symmir -g sar split -noprompt
symrdf -g sar -bcv establish -noprompt
symrdf -g sar -bcv split -noprompt
symmir -g sar establish -full -exact -bcv -rdf -noprompt
symmir -g sar split -bcv -rdf -noprompt
symmir -g sar establish -noprompt
```

If a concurrent BCV setup exists when a SRDF/AR copy cycle begins, SRDF/AR will use the last BCV pair that was established, regardless of whether it was the SRDF/AR BCV pair or the non-SRDF/AR BCV pair. *Prior to starting SRDF/AR, you need to make sure that the last BCV pairs that were established were the SRDF/AR BCV pairs*. In the preceding sequence, the last BCV pairs established were the SRDF/AR BCV pairs. Thus, the required setup sequence has been performed.

1. The ability to access concurrent BCVs while running SDRD/AR begins with Solutions Enabler version 5.2 and Enginuity Version 5568.

- The following `symmir query` command with the `-multi` option examines the relationship and status of the concurrent BCV pairs. The SRDF/AR pairs are devices 012E/04CE and 012F/04CF. The non-SRDF/AR pairs are devices 012E/04CC and 012F/04CD. The non-SRDF/AR pairs are displayed even though they are *not* associated with the device group (indicated by the "N/A" and the absence of the *). All pairs are in the Synchronized state.

```
symmir -g sar query -multi

Device Group (DG) Name: sar
DG's Type             : REGULAR
DG's Symmetrix ID     : 000185500505

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- ----------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001         012E      0 BCV001                 04CE *        0 Synchronized
                         0 N/A                    04CC          0 Synchronized
DEV002         012F      0 BCV002                 04CF *        0 Synchronized
                         0 N/A                    04CD          0 Synchronized

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

Legend:

(*): The paired BCV device is associated with this group.
```

- The `symreplicate` command runs one SRDF/AR copy cycle in the foreground, using the configuration of devices defined in the device group sar and single-hop copy options defined in a file called sar.opt (for file content, refer to the `symrep.opt` file defined in Example 1).

```
symreplicate -g sar -options sar.opt -foreground start -noprompt

Execute a symreplicate 'Start' operation
for device group 'sar' (y/[n]) ? y

Checking for valid group configuration...

Checking for valid initial group state...

Checking for local BCV synchronization...

Splitting local BCV pairs...
```

- While the SRDF/AR copy cycle is in progress, issuing a `symmir query` command with the `-multi` option from a second window indicates that the local SRDF/AR BCV pairs are now split (the preceding `symreplicate` output shows them in process of splitting). The non-SRDF/AR BCV pairs remain in the Synchronized state.

symmir -g sar query -multi

```
Device Group (DG) Name: sar
DG's Type            : REGULAR
DG's Symmetrix ID    : 000185500505

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- -----------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001         012E      0 N/A                    04CC          0 Synchronized
                         0 BCV001                 04CE *       30 Split
DEV002         012F      0 N/A                    04CD          0 Synchronized
                         0 BCV002                 04CF *       30 Split

Total               ------                                -------
  Track(s)               0                                     60
  MB(s)                0.0                                    1.9

Legend:

(*): The paired BCV device is associated with this group.
```

- The `symmir split` command (from the second window) attempts to split the non-SRDF/AR BCV pairs that were defined in the device file named devfile. However, without the `-skip` option, this operation fails because the standard devices are locked as a result of their participation in the SRDF/AR copy cycle.

symmir -f devfile -sid 505 split -instant -noprompt

```
'Split' operation execution is in progress for the device list in device file
   'devfile'. Please wait...

Unable to acquire the Symmetrix device lock
```

- Another attempt to split the non-SRDF/AR BCV pairs uses the `-skip` option and is successful.

symmir -f devfile -sid 505 split -instant -noprompt -skip

```
'Split' operation execution is in progress for the device list in device file
   'devfile'. Please wait...

'Split' operation successfully executed for the device list in device file
   'devfile'.
```

- Another query shows that both the SRDF/AR and non-SRDF/AR BCV pairs are in the Split state.

symmir -g sar query -multi

```
Device Group (DG) Name: sar
DG's Type             : REGULAR
DG's Symmetrix ID     : 000185500505

     Standard Device                        BCV Device                   State
-------------------------- ------------------------------------- -----------
                    Inv.                                     Inv.
Logical         Sym  Tracks Logical                  Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001          012E      0 N/A                      04CC          30 Split
                          0 BCV001                   04CE *        30 Split
DEV002          012F      0 N/A                      04CD          30 Split
                          0 BCV002                   04CF *        30 Split

Total                ------                                   -------
  Track(s)                0                                       120
  MB(s)                 0.0                                       3.8

Legend:

(*): The paired BCV device is associated with this group.
```

Note: The following output in the first window displays the completion of the symreplicate copy cycle that began earlier. Note that the local SRDF/AR BCV pairs are re-established prior to completion.

```
Incrementally establishing RDF pairs...

Waiting for RDF synchronization...

Incrementally establishing local BCV pairs...

Incrementally establishing remote BCV pairs...

Waiting for remote device synchronization...

Splitting remote BCV pairs...

1 cycle(s) complete; exiting symreplicate...
```

Example 6: Accessing Concurrent Non-SRDF/AR BCVs While Running SRDF/AR (Multi-Hop Configuration)[1]

The hardware setup for this multi-hop SRDF/AR configuration consists of two hosts—one connected to a local (source) Symmetrix, and the other connected to a remote (target) Symmetrix (sid 33) at Hop 2. Some commands are issued from the local-site host and some from the remote-site host. The SRDF/AR device group (sar) and the creation of the multi-hop environment have already been set up. The example establishes on the target Symmetrix a set of non-SRDF/AR BCVs that are concurrent with a set of SRDF/AR BCVs there. Devices on the target (Hop 2) Symmetrix are:

- SRDF/AR standard devices 0001–0005
- SRDF/AR BCV devices 0031–0035
- Non-SRDF/AR BCV devices 0043–0047

Although the local-site host has a device *group* defined for running SRDF/AR, you also need to create a device *file* that allows the local-site host to manipulate the SRDF/AR BCV pairs located on the remote Symmetrix array (sid 33) at Hop 2. The following `symmir query` command from the local-site host examines the status of the SRDF/AR BCV pairs on Hop 2 that were defined previously in device file devfile.[2]

`symmir -f devfile query -sid 33`

```
Device File Name      : devfile
Device's Symmetrix ID : 000000005233

     Standard Device                    BCV Device                  State
-------------------------- ------------------------------------- -----------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

N/A            0001      0 N/A                    0031          0 Split
N/A            0002      0 N/A                    0032          0 Split
N/A            0003      0 N/A                    0033          0 Split
N/A            0004      0 N/A                    0034          0 Split
N/A            0005      0 N/A                    0035          0 Split

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

Legend:

(*): The paired BCV device is associated with this group.
```

1. The ability to access concurrent BCVs while running SDRD/AR begins with Solutions Enabler version 5.2 and Enginuity Version 5568.
2. The device file devfile defines the following SRDF/AR pairs:
   0001 0031
   0002 0032
   0003 0033
   0004 0034
   0005 0035

- On the remote-site host, create a device group to manipulate the non-SRDF/AR BCVs. The `symdg create` command creates an R2 type device group (mbcv). The symld addall command adds the SRDF/AR R2 devices to the group, using the `-range` option to limit the selections to those devices between 0001 and 0005. The `symbcv` command associates the non-SRDF/AR BCVs with the device group, using the `-range` option with the `associateall` action to limit the selection to those BCVs that are within the specified range (0043 through 0047).

```
symdg create mbcv -type rdf2
symld -g mbcv addall dev -range 0001:0005 -sid 33
symbcv -g mbcv associateall dev -range 0043:0047
```

- The `symmir establish` command from the remote host fully establishes the BCV pairs in the exact order that they were defined in the device group.

```
symmir -g mbcv establish -full -exact -noprompt

'Full Establish' operation execution is in progress for device group 'mbcv'.
   Please wait...

'Full Establish' operation successfully initiated for device group 'mbcv'.
```

- The `symmir query` command from the remote host displays the status of the non-SRDF/AR BCV pairs (SyncInProg).

```
symmir -g mbcv query

Device Group (DG) Name: mbcv
DG's Type             : RDF2
DG's Symmetrix ID     : 000000005233

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- -----------
                    Inv.                                 Inv.
Logical        Sym  Tracks Logical             Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001         0001      0 BCV001              0043 *     18740 SyncInProg
DEV002         0002      0 BCV002              0044 *      1467 SyncInProg
DEV003         0003      0 BCV003              0045 *      7614 SyncInProg
DEV004         0004      0 BCV004              0046 *      2305 SyncInProg
DEV005         0005      0 BCV005              0047 *     18913 SyncInProg

Total               -------                              -------
  Track(s)                0                                49039
  MB(s)                 0.0                               1532.5

Legend:

(*): The paired BCV device is associated with this group.
```

- The following `symmir query` command from the remote host with the `-multi` option displays the status of the concurrent BCV pairs—both the SRDF/AR BCV pairs and the non-SRDF/AR BCV pairs. The first non-SRDF/AR pair in the display is 0001/0043; the first SRDF/AR pair is 0001/0031. The "N/A" and the absence of an asterisk (*) indicates that the SRDF/AR BCVs are not associated with this device group (mbcv).

`symmir -g mbcv query -multi`

```
Device Group (DG) Name: mbcv
DG's Type             : RDF2
DG's Symmetrix ID     : 000000005233

     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- -----------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001          0001       0 BCV001                0043 *     17027 SyncInProg
                           2 N/A                   0031           0 Split
DEV002          0002       0 BCV002                0044 *         0 Synchronized
                           2 N/A                   0032           0 Split
DEV003          0003       0 BCV003                0045 *      5006 SyncInProg
                           2 N/A                   0033           0 Split
DEV004          0004       0 BCV004                0046 *         0 Synchronized
                           2 N/A                   0034           0 Split
DEV005          0005       0 BCV005                0047 *     17850 SyncInProg
                           2 N/A                   0035           0 Split

Total                 ------                              -------
  Track(s)                10                                39883
  MB(s)                  0.3                               1246.3

Legend:

(*): The paired BCV device is associated with this group.
```

- The `symmir verify` command from the remote host checks the state of the BCV pairs in the device group every five seconds until the non-SRDF/AR BCV pairs are synchronized.

  `symmir -g mbcv verify -i 5`

```
Not all devices in group 'mbcv' are in the 'Synchronized or Restored' state.

Not all devices in group 'mbcv' are in the 'Synchronized or Restored' state.

All devices in group 'mbcv' are in the 'Synchronized or Restored' state.
```

- The `symmir split` command from the remote host splits the non-SRDF/AR BCV pairs.

  `symmir -g mbcv split -noprompt`

```
'Split' operation execution is in progress for device group 'mbcv'.
   Please wait...

'Split' operation successfully executed for device group 'mbcv'.
```

- If a concurrent BCV setup exists when a SRDF/AR copy cycle begins, SRDF/AR will use the last BCV pair that was established and split—regardless of whether it was the SRDF/AR BCV pair or the non-SRDF/AR BCV pair. At this point, the last manipulated BCV pairs were the non-SRDF/AR BCV pairs. The following examples show how to establish and split the SRDF/AR BCV pairs so that they are the last pairs to be manipulated before beginning the symreplicate copy cycle. This manipulation is only required prior to starting SRDF/AR. The following `symmir establish` command from the local-site host establishes the SRDF/AR BCV pairs in device file devfile.

```
symmir -f devfile establish -noprompt -sid 33

'Incremental Establish' operation execution is in progress for the device
   list
   in device file 'devfile'. Please wait...

'Incremental Establish' operation successfully initiated for the device list
   in device file 'devfile'.
```

- The `symmir split` command from the local host splits the SRDF/AR BCV pairs in device file devfile.

```
symmir -f devfile split -nop -sid 33

'Split' operation execution is in progress for the device list in device file
   'devfile'. Please wait...

'Split' operation successfully executed for the device list in device file
   'devfile'.
```

- The `symreplicate` command runs one SRDF/AR copy cycle in the foreground from the local host, using the configuration of devices defined in the device group sar and multi-hop copy options defined in a file called rep.txt (for file content, refer to the `rep_opt.txt` file defined in Example 2).

```
symreplicate -g sar start -options rep.txt -foreground -noprompt

Checking for valid group configuration...

Checking for valid initial group state...

Checking for first hop BCV device synchronization...

Splitting first hop BCV pairs...

Incrementally establishing remote RDF pairs...

Waiting for remote RDF pair synchronization...

Incrementally establishing first hop BCV pairs...

Incrementally establishing second hop BCV pairs...

Waiting for second hop BCV synchronization...

Splitting second hop BCV pairs...
```

- While the SRDF/AR copy cycle is in progress (the preceding `symreplicate` output shows that the SRDF/AR second-hop BCV pairs are in process of splitting), the following `symmir establish` command from the remote-site host attempts to establish the non-SRDF/AR BCV pairs. However, the standard devices are locked as a result of their participation in the SRDF/AR copy cycle, so the operation fails.

```
symmir -g mbcv establish

Execute 'Incremental Establish' operation for device group 'mbcv' (y/[n]) ? y

'Incremental Establish' operation execution is in progress for device group
   'mbcv'. Please wait...

Unable to acquire the Symmetrix device lock
```

- A subsequent `symmir establish` command from the remote host uses the `-skip` option to override the device lock, thus allowing the non-SRDF/AR BCVs to be established with the SRDF/AR standard devices that are participating in the SRDF/AR copy cycle.

```
symmir -g mbcv establish -noprompt -skip

'Incremental Establish' operation execution is in progress for device group
   'mbcv'. Please wait...

'Incremental Establish' operation successfully initiated for device group
   'mbcv'.
```

- The `symmir query` command from the remote host with the `-multi` option displays the status of the concurrent BCV pairs—both the SRDF/AR BCV pairs and the non-SRDF/AR BCV pairs. The SRDF/AR BCV pairs have reached the Split state; the non-SRDF/AR BCV pairs are in the Synchronized state.

```
symmir -g mbcv query -multi

Device Group (DG) Name: mbcv
DG's Type              : RDF2
DG's Symmetrix ID      : 000000005233

     Standard Device                  BCV Device                    State
-------------------------- ------------------------------------- -----------
                    Inv.                                    Inv.
Logical         Sym  Tracks Logical                Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001          0001       0 BCV001                0043 *         0 Synchronized
                           0 N/A                   0031           0 Split
DEV002          0002       0 BCV002                0044 *         0 Synchronized
                           0 N/A                   0032           0 Split
DEV003          0003       0 BCV003                0045 *         0 Synchronized
                           0 N/A                   0033           0 Split
DEV004          0004       0 BCV004                0046 *         0 Synchronized
                           0 N/A                   0034           0 Split
DEV005          0005       0 BCV005                0047 *         0 Synchronized
                           0 N/A                   0035           0 Split

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

Legend:

(*): The paired BCV device is associated with this group.
```

- Because the `symreplicate` copy cycle is still in progress, the following attempt from the remote host to split the non-SRDF/AR BCV pairs without the `-skip` option fails.

`symmir -g mbcv split -noprompt`

```
'Split' operation execution is in progress for device group 'mbcv'. Please
   wait...

Unable to acquire the Symmetrix device lock
```

- Another attempt from the remote host to split the non-SRDF/AR BCV pairs using the `-skip` option succeeds.

`symmir -g mbcv split -noprompt -skip`

```
'Split' operation execution is in progress for device group 'mbcv'. Please
   wait...

'Split' operation successfully executed for device group 'mbcv'.
```

Example 7: Restarting a Replicate Session When Devices Are Locked

Device locks are held during the replicate session to block other applications from altering device states while this session executes. Under certain circumstances, a replicate session may exit with devices left in a locked state. For example, a replicate session may terminate when an RDF link goes down unexpectedly. Then the replicate session cannot restart after the RDF link is brought back up, because of the locked devices. Beginning with EMC Solutions Enabler version 5.2, you can use the `-recover` option with the symreplicate start or restart command to recover the existing device locks and restart the session (SRDF/AR checks if it previously owned the device locks and, if so, proceeds as if it just acquired the existing locks).

Using the `-recover` option allows you to recover without having to manually release the device locks. When SRDF/AR detects a situation where devices are locked and recovery is possible, SRDF/AR returns a message suggesting that you attempt to recover.

CAUTION

Caution: Before using the –recover option, make sure no other replicate session that uses the same device group is currently running.

- The following `symreplicate restart` command attempts to restart a replicate session involving devices in the device group sar2. However, the output indicates that SRDF/AR is unable to do so at this time, because it cannot "lock the local devices," indicating that the devices in sar2 are already in a locked state.

`symreplicate restart -g sar2 -foreground -noprompt`

```
Checking for valid group configuration...

Checking for valid initial group state...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices; waiting for retry...

Can't lock local devices.

Unable to acquire the Symmetrix device lock

If you are sure no other symreplicate process is currently active for group
'sar2', the locks can be recovered by specifying the '-recover' option.
```

The "If you are sure" message above will not be displayed if SRDF/AR detects that the locks cannot be recovered, or that the base daemon is running. If the base daemon is running, the device locks will be released eventually. You can wait a short time and retry the operation. If the base daemon is not running, you can release the device locks manually (for example, `symdg -lock release sar2`).

- The `symreplicate restart` command is repeated here using the `-recover` option. SRDF/AR resolves the locked device situation and is able to restart the replicate session normally.

  `symreplicate restart -g sar2 -foreground -noprompt -recover`

```
Checking for valid group configuration...

Checking for valid initial group state...

Checking for first hop BCV device synchronization...

Splitting first hop BCV pairs...

Incrementally establishing remote RDF pairs...

Waiting for remote RDF pair synchronization...

Incrementally establishing first hop BCV pairs...

Incrementally establishing second hop BCV pairs...

Waiting for second hop BCV synchronization...

Splitting second hop BCV pairs...

1 cycle(s) complete; exiting symreplicate...
```

9

Querying and Verifying with SRDF Commands

This chapter provides examples of the Symmetrix command line interface (SYMCLI) actions and specific commands, which are used to query and verify SRDF group operations. It focuses on the various arguments, options, and the application of certain parameters for the SRDF query and verify actions. Using examples of SRDF commands, it describes how to manage the behavior and states of the various SRDF components in a typical configuration.

Note: Some of the examples in this section were performed with earlier versions of software. Therefore, your output displays may not look exactly like the ones appearing in these examples.

Example 1: Querying a Device Group

Before creating a device group and adding devices to it, examine the devices on your local Symmetrix to determine which are source devices (Sym Dev), which are remote target devices (RDev), and whether a device is an R1 or R2 type device. The `symrdf list` command displays this information as well as other relevant data such as RDF group (G), replication method (column M), pair state, invalid tracks, and the state of each device and the RDF links that connect them. The ellipsis (…) represents truncated output.

```
symrdf list

Symmetrix ID: 000000003264

                              Local Device View
---------------------------------------------------------------------------
                   STATUS    MODES                      RDF  S T A T E S
Sym        RDF    --------- -----  R1 Inv   R2 Inv ----------------------
Dev  RDev  Typ:G  SA RA LNK  MDA   Tracks   Tracks Dev RDev Pair
---- ---- ------ --------- ----- ------- ------- --- ---- -------------

0045 0045   R2:2  RW WD NR   S..         0   49500 WD  RW   Suspended
0046 0046   R2:2  ?? WD NR   S..         0   33000 WD  RW   Suspended
0047 0047   R2:2  ?? WD NR   S..         0       0 WD  RW   Suspended
009C 0054   R1:2  RW RW RW   S..         0       0 RW  NR   Synchronized
009D 0055   R1:2  RW RW RW   S..         0       0 RW  NR   Synchronized
009E 0056   R1:2  RW RW RW   S..         0       0 RW  NR   Synchronized
009F 0057   R1:2  RW RW RW   S..         0       0 RW  NR   Synchronized
00A0 0058   R1:2  RW RW RW   A.W         0       0 RW  NR   Synchronized
00A1 0059   R1:2  RW RW RW   A.W         0       0 RW  NR   Synchronized
00A2 005A   R1:2  RW RW RW   A.W         0       0 RW  NR   Synchronized
00A3 005B   R1:2  RW RW RW   A.W         0       0 RW  NR   Synchronized
```

- The `symdev list` command with the `-r1` option displays all R1 devices. Those R1 devices that are not already part of a device group are displayed as “N/Grp’d,” which means they are available to be added to a new RDF1 device group.

```
symdev list -r1

Symmetrix ID: 000000003264

        Device Name           Directors                  Device
---------------------------- ------------ ----------------------------------
                                                        Cap
Sym  Physical                 SA :P DA :IT Config        Attribute    Sts  (MB)
---------------------------- ------------ ----------------------------------

009C /dev/rdsk/emcpower84c    16B:1 01A:C0 RDF1          N/Grp'd      RW  516
009D /dev/rdsk/emcpower85c    16B:1 02B:D3 RDF1          N/Grp'd      RW  516
009E /dev/rdsk/emcpower90c    16B:1 02A:C0 RDF1          N/Grp'd      RW  516
009F /dev/rdsk/emcpower91c    16B:1 01B:D3 RDF1          N/Grp'd      RW  516
00A0 /dev/rdsk/emcpower92c    16B:1 01B:C0 RDF1          N/Grp'd      RW  516
00A1 /dev/rdsk/emcpower93c    16B:1 02A:D3 RDF1          Grp'd        RW  516
00A2 /dev/rdsk/emcpower94c    16B:1 02B:C0 RDF1          N/Grp'd      RW  516
```

- Creating a device group and adding devices to it are prerequisites for performing SRDF operations. The `symdg create` command creates a device group (Rdf1Grp). The `symld add` commands add standard devices to the group, using either a device's physical device (pd) name or, as shown below, its Symmetrix device (dev) name.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp -sid 3264 add dev 09C
symld -g Rdf1Grp -sid 3264 add dev 09D
symdg show Rdf1Grp

Group Name:  Rdf1Grp

    Group Type                                   : RDF1
    Device Group in GNS                          : Yes
    Valid                                        : Yes
    Symmetrix ID                                 : 000000003264
    Group Creation Time                          : Tue Jan  6 12:08:17 2004
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :    2
    Number of Associated GK's                    :    0
    Number of Locally-associated BCV's           :    0
    Number of Locally-associated VDEV's          :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (2):
        {
        --------------------------------------------------------------------
                                                       Sym                Cap
        LdevName              PdevName                 Dev  Att. Sts      (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c2t6d3s2       009C      RW        516
        DEV002                /dev/rdsk/c2t6d4s2       009D      RW        516
        }

    Device Group RDF Information
        {
        RDF Type                                   : R1
        RDF (RA) Group Number                      : 2                 (01)

        Remote Symmetrix ID                        : 000000003265

        R2 Device Is Larger Than The R1 Device     : False

        RDF Mode                                   : Synchronous
        RDF Adaptive Copy                          : Disabled
        RDF Adaptive Copy Write Pending State      : N/A
        RDF Adaptive Copy Skew (Tracks)            : 65535

        RDF Device Domino                          : Disabled

        RDF Link Configuration                     : Fibre
        RDF Link Domino                            : Disabled
        Prevent Automatic RDF Link Recovery        : Disabled
        Prevent RAs Online Upon Power ON           : Enabled

        Device RDF Status                          : Ready             (RW)

        Device RA Status                           : Ready             (RW)
        Device Link Status                         : Ready             (RW)

        Device Suspend State                       : N/A
        Device Consistency State                   : Disabled
```

```
RDF R2 Not Ready If Invalid               : Enabled

Device RDF State                          : Ready           (RW)
Remote Device RDF State                   : Not Ready       (NR)

RDF Pair State (  R1 <===> R2 )           : Synchronized

Number of R1 Invalid Tracks               : 0
Number of R2 Invalid Tracks               : 0
}
```

- When EMC installs an SRDF configuration, the installers usually establish static SRDF pairs at that time. The `symrdf query` command demonstrates the state of the SRDF devices and their RDF links. Under normal circumstances, the SRDF pair is synchronized (as shown below). The R1 devices are read-writeable and the RDF links are read-writeable. However, the R2 devices, which are acting as mirrors to the R1 devices, are write disabled (WD) and cannot be written to by the target-side host at this time. The link is operating in Synchronous replication (indicated by an S in the M column).

symrdf -g Rdf1Grp query

```
Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
            ST                  LI      ST
Standard     A                   N       A
Logical      T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002  009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized

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

- The following `symrdf` command splits the SRDF pairs in the device group. As the `split` occurs, the singular SRDF control operations `suspend` and `rw_enable r2` occur. When the `split` is complete, a `query` operation will reveal the altered state of the links and the R2 devices.

symrdf -g Rdf1Grp -noprompt split

```
An RDF 'Split' operation execution is in progress for device group 'Rdf1Grp'.
   Please wait...

     Suspend RDF link(s).......................................Done.
     Read/Write Enable device(s) on RA at target (R2)..........Done.

The RDF 'Split' operation successfully executed for device group 'Rdf1Grp'.
```

◆ The following query operation reveals that the links have been logically set to NR (not ready) and the R2 device state has changed from WD to RW.

symrdf query

```
Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 NR 0054 RW       0        0 S..   Split
DEV002  009D RW       0        0 NR 0055 RW       0        0 S..   Split

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

◆ The following `symrdf` command performs an incremental establish for the SRDF pairs in device group Rdf1Grp. The operation copies to the R2 devices any changes that have been made to the R1 devices while the devices were split. Like all SRDF control operations, you can initiate the establish action from either the source or target side with the same results. The individual operations that combine to create an establish action are logged as they occur. For a more detailed report, you can examine the log file in /var/symapi/log/symapi-yyyymmdd.log.

symrdf -g Rdf1Grp -noprompt establish

```
An RDF 'Incremental Establish' operation execution is in progress for device
   group 'Rdf1Grp'. Please wait...

    Write Disable device(s) on RA at target (R2)..............Done.
    Suspend RDF link(s).......................................Done.
    Mark target (R2) devices to refresh from source (R1)......Started.
    Device: 0054 ............................................. Marked.
    Mark target (R2) devices to refresh from source (R1)......Done.
    Suspend RDF link(s).......................................Done.
    Merge device track tables between source and target.......Started.
    Device: 009C ............................................. Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s)........................................Done.

The RDF 'Incremental Establish' operation successfully initiated for device
   group 'Rdf1Grp'.
```

- An immediate `symrdf verify` command indicates that neither of the SRDF pairs is synchronized. The `echo $status` value of 5 is the code number that indicates no devices are synchronized. Because the devices are in the process of synchronizing, verifying the SyncInProg state will return a zero value that indicates success.

```
symrdf -g Rdf1Grp verify -synchronized

None of the devices in the RDF group 'Rdf1Grp' are in the 'Synchronized'
   state.

echo $status
5

symrdf -g Rdf1Grp verify -syncinprog

All devices in the RDF group ' Rdf1Grp' are in the 'SyncInProg' state.

echo $status
0
```

- After some time elapses, the `symrdf query` command displays that one of the SRDF pairs is fully synchronized and one is still in the process of synchronizing. As of this query snapshot, 830 remote (R2) invalid tracks on the source (R1) side still remain to be copied to the target device (055) to complete the synchronization process. The remote (R2) invalid tracks on the R1 side represent those tracks that are still “owed” to the R2 side.

```
symrdf -g Rdf1Grp query

Device Group (DG) Name: Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 NR       0        0 S..   Split
DEV002  009D RW       0        0 RW 0055 NR       0        0 S..   Split

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

- The `symrdf verify` command displays a message every 30 seconds until both SRDF pairs in the group are synchronized.

```
symrdf -g Rdf1Grp verify -i 30 -synchronized

Not all devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.

Not all devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.

All devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.
```

- Examine the return codes from the following `symrdf verify` commands. While `verify` and `verify -synchronized` return the success code 0, attempting to verify other states returns the appropriate failure code.

```
symrdf -g Rdf1Grp verify

All devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.

echo $status
0

symrdf -g Rdf1Grp -synchronized verify

All devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.

echo $status
0

symrdf -g Rdf1Grp -failedover verify

None of the devices in the RDF group 'Rdf1Grp' are in the 'Failed Over'
   state.

echo $status
34

symrdf -g Rdf1Grp -syncinprog verify

None of the devices in the RDF group 'Rdf1Grp' are in the 'SyncInProg'
   state.

echo $status
28

symrdf -g Rdf1Grp -split verify

None of the devices in the RDF group 'Rdf1Grp' are in the 'Split' state.

echo $status
26
```

- Both SRDF pairs in the device group are fully synchronized. The following `symrdf split` commands split the SRDF pairs in stages to show the values returned by the `symrdf verify -split` commands.

```
symrdf -g Rdf1Grp split DEV001 -noprompt

An RDF 'Split' operation execution is in progress for device 'DEV001' in
   device group 'Rdf1Grp'. Please wait...

The RDF 'Split' operation successfully executed for device 'DEV001' in
   device group 'Rdf1Grp'.

symrdf -g Rdf1Grp -split verify

Not all devices in the RDF group 'Rdf1Grp' are in the 'Split' state.

echo $status
25
```

- Once the previous split operation has completed, the following `symrdf split` command can successfully split the other SRDF pair in the group. The subsequent `symrdf verify -split` command returns the success value (zero).

```
symrdf -g Rdf1Grp split -noprompt

An RDF 'Split' operation execution is in progress for device group 'Rdf1Grp'.
   Please wait...

The RDF 'Split' operation successfully executed for device group 'Rdf1Grp'.

symrdf -g Rdf1Grp -split verify

All devices in the RDF group 'Rdf1Grp' are in the 'Split' state.

echo $status
0
```

- When you initiate an SRDF control operation, the system checks the state of each SRDF pair involved in the operation. If a pair is not in an SRDF pair state that is valid (legal) for that operation, the operation will fail unless the `-force` option is used with the command. The following command without `-force` rejects the failover operation because the SRDF pairs are currently in the Split state, which is not a legal state for failover.

```
symrdf -g Rdf1Grp -noprompt failover

An RDF 'Failover' operation execution is in progress for device group
   'Rdf1Grp'.
   Please wait...

Cannot proceed because the device pair is not in a legal RDF state.
```

- The following command with the `-force` option forces the failover operation to occur despite the unexpected Split state.

```
symrdf -g Rdf1Grp -noprompt -force failover

An RDF 'Failover' operation execution is in progress for device group
   'Rdf1Grp'.
   Please wait...

     Write Disable device(s) on SA at source (R1)..............Done.
     Suspend RDF link(s).......................................Done.
     Read/Write Enable device(s) on RA at target (R2)..........Done.

The RDF 'Failover' operation successfully executed for device group
   'Rdf1Grp'.
```

- The following `query` operation displays the results of the failover operation. The R1 devices in each SRDF pair are write disabled (WD), the RDF links are suspended (NR), and the R2 devices are read/write enabled (RW).

symrdf -g Rdf1Grp query

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C WD       0        0 NR 0054 RW       0        0 S..   Failed Over
DEV002  009D WD       0        0 NR 0055 RW       0        0 S..   Failed Over

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

- The `symrdf failback` command initiates a failback on one of the two SRDF pairs in the device group.

symrdf -g Rdf1Grp failback DEV001 -noprompt

```
An RDF 'Failback' operation execution is in progress for device 'DEV001' in
   group 'Rdf1Grp'. Please wait...

    Write Disable device(s) on RA at target (R2).............Done.
    Suspend RDF link(s)......................................Done.
    Merge device track tables between source and target.......Started.
    Device: 009C ............................................ Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s).......................................Done.
    Read/Write Enable device(s) on SA at source (R1).........Done.

The RDF 'Failback' operation successfully executed for device 'DEV001' in
   group 'Rdf1Grp'.
```

- The `symrdf query` command displays the states of the SRDF pairs in the device group. The two SRDF pairs are now in different states — one pair is in the Synchronized state and the other pair is still in the Failed Over state.

`symrdf -g Rdf1Grp query`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S.. Synchronized
DEV002  009D WD       0        0 NR 0055 RW       0        0 S..  Failed Over

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

- This subsequent `symrdf failback` command initiates a failback on the SRDF pair that is still in the Failed Over state.

`symrdf -g Rdf1Grp failback DEV002 -noprompt`

```
An RDF 'Failback' operation execution is in progress for device 'DEV002' in
   group 'Rdf1Grp'. Please wait...

    Write Disable device(s) on RA at target (R2).............Done.
    Suspend RDF link(s).....................................Done.
    Merge device track tables between source and target.......Started.
    Device: 009D ........................................... Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s)......................................Done.
    Read/Write Enable device(s) on SA at source (R1).........Done.

The RDF 'Failback' operation successfully executed for device 'DEV002' in
   group 'Rdf1Grp'.
```

- The `symrdf query` command displays again the states of the SRDF pairs in the device group. The two SRDF pairs are now in complementary states — one pair is in the Synchronized state and the other pair is in the SyncInProg state.

```
symrdf -g Rdf1Grp query

Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S.. Synchronized
DEV002  009D RW    1048        0 RW 0055 WD    1048        0 S..   SyncInProg

Total           -------- --------           -------- --------
  Track(s)          1048        0               1048        0
  MB(s)             32.0      0.0               32.0      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
```

- Because there are different SRDF pair states in the device group, the following `symdg show` command displays that the composite state of SRDF pairs in the group is Mixed.

```
symdg show Rdf1Grp

Group Name:  Rdf1Grp

    Group Type                                      : RDF1
    Device Group in GNS                             : Yes
    Valid                                           : Yes
    Symmetrix ID                                    : 000000003264
    Group Creation Time                             : Tue Jan  6 12:31:23 2004
    Vendor ID                                       : EMC Corp
    Application ID                                  : SYMCLI

    Number of STD Devices in Group                  :    2
    Number of Associated GK's                       :    0
    Number of Locally-associated BCV's              :    0
    Number of Locally-associated VDEV's             :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (2):
        {
        --------------------------------------------------------------------
                                                      Sym                Cap
        LdevName              PdevName                Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c2t6d3s2      009C      RW       516
        DEV002                /dev/rdsk/c2t6d4s2      009D      RW       516
        }

   Device Group RDF Information
        {
        RDF Type                                  : R1
        RDF (RA) Group Number                     : 2               (01)

        Remote Symmetrix ID                       : 000000003265

        R2 Device Is Larger Than The R1 Device : False

        RDF Mode                                  : Synchronous
        RDF Adaptive Copy                         : Disabled
        RDF Adaptive Copy Write Pending State  : N/A
        RDF Adaptive Copy Skew (Tracks)           : 65535

        RDF Device Domino                         : Disabled

        RDF Link Configuration                    : Fibre
        RDF Link Domino                           : Disabled
        Prevent Automatic RDF Link Recovery       : Disabled
        Prevent RAs Online Upon Power ON          : Enabled

        Device RDF Status                         : Ready           (RW)

        Device RA Status                          : Ready           (RW)
        Device Link Status                        : Ready           (RW)

        Device Suspend State                      : N/A
        Device Consistency State                  : Disabled
        RDF R2 Not Ready If Invalid               : Enabled

        Device RDF State                          : Ready           (RW)
        Remote Device RDF State                   : Not Ready       (NR)

        RDF Pair State (  M I X E D   )           : Mixed
```

```
        Number of R1 Invalid Tracks             :        0
        Number of R2 Invalid Tracks             :     1048
        }
```

- As a prerequisite for associating RDF1 BCV devices with the device group, the `symdev list` command with the `-r1 -bcv` options displays all RDF1 BCV devices. Those that are not already part of a device group (N/Asst'd) are free to be added to device group Rdf1Grp.

`symdev list -r1 -bcv`

```
Symmetrix ID: 000000003264

        Device Name           Directors                   Device
---------------------------- ------------ ----------------------------------
                                                        Cap
Sym  Physical                SA :P DA :IT Config        Attribute Sts  (MB)
---------------------------- ------------ ----------------------------------

00A6 /dev/rdsk/emcpower98c   16B:1 02A:D0 RDF1-BCV      N/Asst'd     RW516
00A7 /dev/rdsk/emcpower99c   16B:1 01B:C3 RDF1-BCV      N/Asst'd     RW516
00A8 /dev/rdsk/emcpower100c  16B:1 01B:D0 RDF1-BCV      N/Asst'd     RW516
00A9 /dev/rdsk/emcpower101c  16B:1 02A:C3 RDF1-BCV      N/Asst'd     RW516
00AA /dev/rdsk/emcpower121c  16B:1 02B:D0 RDF1-BCV      N/Asst'd     RW516
00AB /dev/rdsk/emcpower122c  16B:1 01A:C3 RDF1-BCV      N/Asst'd     RW516
00AC /dev/rdsk/emcpower123c  16B:1 01B:C1 RDF1-BCV      N/Asst'd     RW516
00AD /dev/rdsk/emcpower124c  16B:1 02A:D2 RDF1-BCV      N/Asst'd     RW516
00AE /dev/rdsk/emcpower125c  16B:1 02B:C1 RDF1-BCV      N/Asst'd     RW516
00AF /dev/rdsk/emcpower126c  16B:1 01A:D2 RDF1-BCV      N/Asst'd     RW516
```

- The following `symbcv` commands associate two of these RDF1 BCV devices (A6 and A7) with device group Rdf1Grp. SYMCLI assigns the devices the default logical names BCV001 and BCV002, respectively.

  `symbcv -g Rdf1Grp -sid 3264 associate dev A6`
  `symbcv -g Rdf1Grp -sid 3264 associate dev A7`

- The `symrdf query` command without options displays only the SRDF pair state of the RDF standard devices in the device group.

`symrdf -g Rdf1Grp query`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI       ST
Standard      A                    N        A
Logical       T  R1 Inv   R2 Inv  K         T  R1 Inv   R2 Inv       RDF Pair
Device   Dev  E  Tracks   Tracks  S Dev     E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001   009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002   009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized

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

- The `symrdf query` command with the `-bcv` option displays only the SRDF pair state of the RDF BCV devices in the device group.

`symrdf -g Rdf1Grp query -bcv`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

                         R E M O T E     S Y M M E T R I X

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
BCV           A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA    STATE
-------------------------------- -- ------------------------ ----- ---------

BCV001  00A6 RW       0        0 RW 005E WD       0        0 S..   Synchronized
BCV002  00A7 RW       0        0 RW 005F WD       0        0 S..   Synchronized

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

- The `symrdf query` command with the `-all` option displays the SRDF pair state of all devices in the device group, regardless of device type.

`symrdf -g Rdf1Grp query -all`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI       ST
Standard      A                   N        A
Logical       T  R1 Inv   R2 Inv  K        T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev    E  Tracks   Tracks MDA    STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002  009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized
BCV001  00A6 RW       0        0 RW 005E WD       0        0 S..   Synchronized
BCV002  00A7 RW       0        0 RW 005F WD       0        0 S..   Synchronized

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

- The `symmir establish` command creates BCV pairs. The `-exact` option matches standard devices with BCV devices in the exact order that they were added to the device group. Thus, device 9C will be established with device A6, and 9D with A7. To perform `symmir` commands requires that you have TimeFinder™/Mirror software.

`symmir -g Rdf1Grp establish -full -exact -noprompt`

```
'Full Establish' operation execution is in progress for device group
   'Rdf1Grp'.
   Please wait...

'Full Establish' operation successfully initiated for device group 'Rdf1Grp'.
```

- The `symmir query` command displays the BCV pairs in the device group and their state of mirroring. Both BCV pairs are in the process of synchronizing.

`symmir -g Rdf1Grp query`

```
Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                 State
-------------------------- ------------------------------------- -----------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical              Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------

DEV001          009C      0 BCV001              00A6 *     10554 SyncInProg
DEV002          009D      0 BCV002              00A7 *     10469 SyncInProg

Total                -------                              -------
  Track(s)                 0                                21023
  MB(s)                  0.0                                657.0

Legend:

(*): The paired BCV device is associated with this group.
```

- The `symrdf query -all` command displays the SRDF pair state of all RDF devices in the device group. While established with DEV001 and DEV002 (i.e., as part of a BCV pair), the RDF1 BCV devices are in a Suspended SRDF pair state and cannot copy data to their respective target devices (05E and 05F).

`symrdf -g Rdf1Grp query -all`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                   LI      ST
Standard      A                    N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
------------------------------- -- ------------------------ ----- ---------

DEV001  009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002  009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized
BCV001  00A6 NR       0        0 NR 005E WD       0        0 S..   Suspended
BCV002  00A7 NR       0        0 NR 005F WD       0        0 S..   Suspended

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

- The `symdg show` command displays group information about device group Rdf1Grp. The group contains two RDF1 standard devices and two RDF1 BCV devices. The "Device Group RDF Information" section of the display shows that the composite SRDF pair state of the RDF1 standard devices is Synchronized. The "Device Group BCV RDF Information" section of the display shows that the composite SRDF pair state of the RDF1 BCV devices is Suspended.

symdg show Rdf1Grp

```
Group Name:  Rdf1Grp

    Group Type                                    : RDF1
    Device Group in GNS                           : Yes
    Valid                                         : Yes
    Symmetrix ID                                  : 000000003264
    Group Creation Time                           : Tue Jan  6 12:48:27 2004
    Vendor ID                                     : EMC Corp
    Application ID                                : SYMCLI

    Number of STD Devices in Group                :    2
    Number of Associated GK's                     :    0
    Number of Locally-associated BCV's            :    2
    Number of Locally-associated VDEV's           :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (2):
        {
        --------------------------------------------------------------------
                                                    Sym               Cap
        LdevName              PdevName              Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c2t6d3s2    009C      RW       516
        DEV002                /dev/rdsk/c2t6d4s2    009D      RW       516
        }

    BCV Devices Locally-associated (2):
        {
        --------------------------------------------------------------------
                                                    Sym               Cap
        LdevName              PdevName              Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        BCV001                /dev/rdsk/emcpower98c 00A6      RW       516
        BCV002                /dev/rdsk/emcpower99c 00A7      RW       516
        }

    Device Group RDF Information
        {
        RDF Type                                : R1
        RDF (RA) Group Number                   : 2                (01)

        Remote Symmetrix ID                     : 000000003265

        R2 Device Is Larger Than The R1 Device : False

        RDF Mode                                : Synchronous
        RDF Adaptive Copy                       : Disabled
        RDF Adaptive Copy Write Pending State   : N/A
        RDF Adaptive Copy Skew (Tracks)         : 65535

        RDF Device Domino                       : Disabled

        RDF Link Configuration                  : Fibre
        RDF Link Domino                         : Disabled
        Prevent Automatic RDF Link Recovery     : Disabled
```

```
    Prevent RAs Online Upon Power ON       : Enabled

    Device RDF Status                      : Ready          (RW)

    Device RA Status                       : Ready          (RW)
    Device Link Status                     : Ready          (RW)

    Device Suspend State                   : N/A
    Device Consistency State               : Disabled
    RDF R2 Not Ready If Invalid            : Enabled

    Device RDF State                       : Ready          (RW)
    Remote Device RDF State                : Write Disabled (WD)

    RDF Pair State (  R1 <===> R2 )        : Synchronized

    Number of R1 Invalid Tracks            : 0
    Number of R2 Invalid Tracks            : 0
    }

Device Group BCV RDF Information
    {
    RDF Type                               : R1
    RDF (RA) Group Number                  : 2              (01)

    Remote Symmetrix ID                    : 000000003265

    R2 Device Is Larger Than The R1 Device : False

    RDF Mode                               : Synchronous
    RDF Adaptive Copy                      : Disabled
    RDF Adaptive Copy Write Pending State  : N/A
    RDF Adaptive Copy Skew (Tracks)        : 65535

    RDF Device Domino                      : Disabled

    RDF Link Configuration                 : Fibre
    RDF Link Domino                        : Disabled
    Prevent Automatic RDF Link Recovery    : Disabled
    Prevent RAs Online Upon Power ON       : Enabled

    Device RDF Status                      : Ready          (RW)

    Device RA Status                       : Ready          (RW)
    Device Link Status                     : Not Ready      (NR)

    Device Suspend State                   : N/A
    Device Consistency State               : Disabled
    RDF R2 Not Ready If Invalid            : Enabled

    Device RDF State                       : Ready          (RW)
    Remote Device RDF State                : Write Disabled (WD)

    RDF Pair State ( R1 <-  -> R2 )        : Suspended

    Number of R1 Invalid Tracks            : 0
    Number of R2 Invalid Tracks            : 0
    }
```

- The `symmir split` command splits the BCV pairs in the device group Rdf1Grp. When the split completes, the RDF links for the RDF1 BCV devices will still be not ready (NR), even though the state of the source (R1) RDF1 BCV devices will be changed from not ready to read/write enabled. For more information on TimeFinder/Mirror splits, refer to the *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

  `symmir -g Rdf1Grp split -noprompt`

  ```
  'Split' operation execution is in progress for device group 'Rdf1Grp'.
     Please wait...

  'Split' operation successfully executed for device group 'Rdf1Grp'.
  ```

- The `symrdf query -bcv` command shows the SRDF pair state of the RDF1 BCV devices in the device group. Because the links for those devices are not ready (read/write disabled), the SRDF pair state remains Suspended. On the source (R1) side, each RDF1 BCV device has 16500 remote (R2) invalid tracks that need to be copied to the BCV's remote (R2) mirror when synchronization begins.

`symrdf -g Rdf1Grp query -bcv`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

                     R E M O T E     S Y M M E T R I X

       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
BCV           A                   N       A
Logical       T  R1 Inv   R2 Inv K        T  R1 Inv   R2 Inv       RDF Pair
Device   Dev  E  Tracks   Tracks S Dev    E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

BCV001   00A6 RW      0    16500 NR 005E WD       0        0 S..   Suspended
BCV002   00A7 RW      0    16500 NR 005F WD       0        0 S..   Suspended

Total           -------  -------            -------  -------
  Track(s)            0    33000                  0        0
  MB(s)             0.0   1031.0                0.0      0.0
```

- The `symrdf establish` command with the `-bcv` option resumes the RDF links for the RDF1 BCV devices and initiates the propagation of data from the source (R1) RDF1 BCV devices to their remote (R2) mirror devices.

`symrdf -g Rdf1Grp establish -bcv -noprompt`

```
An RDF 'Incremental Establish' operation execution is in progress for device
   group 'Rdf1Grp'. Please wait...

    Suspend RDF link(s).......................................Done.
    Resume RDF link(s)........................................Done.

The RDF 'Incremental Establish' operation successfully initiated for device
   group 'Rdf1Grp'.
```

- The `symrdf query -bcv` command shows again the SRDF pair state of the RDF1 BCV devices (now SyncInProg) and the number of remote (R2) invalid tracks on the source (R1) side that still need to be copied to the BCVs' remote (R2) mirrors to complete the synchronization process. Note, too, that the establish operation changed the state of the links from NR (not ready) to RW (read/write enabled).

`symrdf -g Rdf1Grp query -bcv`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
BCV           A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device   Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

BCV001   00A6 RW       0    13558 RW 005E WD       0        0 S..   SyncInProg
BCV002   00A7 RW       0    16500 RW 005F WD       0        0 S..   SyncInProg

Total           -------  -------            -------  -------
  Track(s)            0    30058                  0        0
  MB(s)             0.0    939.0                0.0      0.0
```

- The `symrdf verify -all` command checks the state of all SRDF pairs in the device group every five seconds until all SRDF pairs are synchronized. Then the verify loop ends.

  `symrdf -g Rdf1Grp verify -all -i 5 -synchronized`

```
NOT all of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

All devices in the RDF group 'Rdf1Grp' are in the 'Synchronized' state.
```

- The `symrdf query -all` command displays all SRDF devices and their states. Like the RDF1 standard devices, the RDF1 BCV devices are now in the Synchronized SRDF pair state. The copying of data from the source (R1) side to the target (R2) side is complete.

`symrdf -g Rdf1Grp query -all`

```
Device Group (DG) Name : Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000000003264

       Source (R1) View                 Target (R2) View     MODES
--------------------------------    ------------------------ ----- ----------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device   Dev  E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ---------

DEV001   009C RW       0        0 RW 0054 WD       0        0 S..   Synchronized
DEV002   009D RW       0        0 RW 0055 WD       0        0 S..   Synchronized
BCV001   00A6 RW       0        0 RW 005E WD       0        0 S..   Synchronized
BCV002   00A7 RW       0        0 RW 005F WD       0        0 S..   Synchronized

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

Example 2: Querying a Composite Group

Querying a composite group is similar to querying a device group, except that the `symrdf query` command includes the `-cg` option and the name of the composite group. This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host connected to two source Symmetrix units (Symmetrix 000187900035 and Symmetrix 000000003143). The example builds a composite group with source R1 devices from both Symmetrix units and enables consistency protection for the composite group. For more examples using SRDF consistency protection, refer to *RDF Consistency Group Operations* on page 3-46.

- The `symcg create` command creates an RDF1 type composite group named SRDF on this host. If you intend to enable the group for consistency protection and have not set the `SYMAPI_RDF_CG_TO_PPATH` variable to `ENABLE`, you must include the `-ppath` option so that the group is added to PowerPath.

  `symcg create SRDF -type rdf1 -ppath`

- The following `symcg addall` command adds to the composite group a range of PowerPath standard devices from Symmetrix 000187900035.

  `symcg -cg SRDF addall dev -range 137:14F -sid 35`

- The following `symcg addall` command adds to the composite group a range of PowerPath standard devices from Symmetrix 000000003143.

  `symcg -cg SRDF addall dev -range F7:10F -sid 43`

- The `symrdf query` command checks the state of the SRDF pairs. Note that SRDF pairs from one Symmetrix unit are in the Suspended state, while the other Symmetrix unit has synchronized SRDF pairs.

`symrdf -cg SRDF query`

```
Composite Group Name     :  SRDF
Composite Group Type     :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID        : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number      :  1 (A)

   Source (R1) View         Target (R2) View       MODES  STATES
----------------------- ------------------------- ----- ------ ------------
     ST                  LI       ST                      C   S
      A                   N        A                      o   u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv      n   s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA  s   p   STATE
----------------------- -- ----------------------- ----- ------ ------------
00F7 RW       0      46 NR 0062 NR       0       0 S..   .  -   Suspended
00FA RW       0      46 NR 0065 NR       0       0 S..   .  -   Suspended
00FC RW       0      46 NR 0067 NR       0       0 S..   .  -   Suspended
00FD RW       0      46 NR 0068 NR       0       0 S..   .  -   Suspended
00FE RW       0      46 NR 0069 NR       0       0 S..   .  -   Suspended
00FF RW       0      46 NR 006A NR       0       0 S..   .  -   Suspended
0100 RW       0      46 NR 006B NR       0       0 S..   .  -   Suspended
0101 RW       0      46 NR 006C NR       0       0 S..   .  -   Suspended
0102 RW       0      46 NR 006D NR       0       0 S..   .  -   Suspended
0103 RW       0      46 NR 006E NR       0       0 S..   .  -   Suspended
0104 RW       0      46 NR 006F NR       0       0 S..   .  -   Suspended
0105 RW       0      46 NR 0070 NR       0       0 S..   .  -   Suspended
0106 RW       0      46 NR 0071 NR       0       0 S..   .  -   Suspended
0107 RW       0      46 NR 0072 NR       0       0 S..   .  -   Suspended
```

```
0108 RW       0       46 NR 0073 NR       0        0 S..    .  -   Suspended
0109 RW       0       46 NR 0074 NR       0        0 S..    .  -   Suspended
010A RW       0       46 NR 0075 NR       0        0 S..    .  -   Suspended
010B RW       0       46 NR 0076 NR       0        0 S..    .  -   Suspended
010C RW       0       46 NR 0077 NR       0        0 S..    .  -   Suspended
010D RW       0       46 NR 0078 NR       0        0 S..    .  -   Suspended
010E RW       0       46 NR 0079 NR       0        0 S..    .  -   Suspended
010F RW       0       46 NR 007A NR       0        0 S..    .  -   Suspended

Symmetrix ID                : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID         : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI       ST                     C   S
      A                   N        A                     o   u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv     n   s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA s   p   STATE
----------------------- -- ------------------------ ----- ------ ------------
0137 RW       0        0 RW 0056 NR       0        0 S..    .  -   Synchronized
013A RW       0        0 RW 0059 NR       0        0 S..    .  -   Synchronized
013C RW       0        0 RW 005B NR       0        0 S..    .  -   Synchronized
013D RW       0        0 RW 005C NR       0        0 S..    .  -   Synchronized
013E RW       0        0 RW 005D NR       0        0 S..    .  -   Synchronized
013F RW       0        0 RW 005E NR       0        0 S..    .  -   Synchronized
0140 RW       0        0 RW 005F NR       0        0 S..    .  -   Synchronized
0141 RW       0        0 RW 0060 NR       0        0 S..    .  -   Synchronized
0142 RW       0        0 RW 0061 NR       0        0 S..    .  -   Synchronized
0143 RW       0        0 RW 0062 NR       0        0 S..    .  -   Synchronized
0144 RW       0        0 RW 0063 NR       0        0 S..    .  -   Synchronized
0145 RW       0        0 RW 0064 NR       0        0 S..    .  -   Synchronized
0146 RW       0        0 RW 0065 NR       0        0 S..    .  -   Synchronized
0147 RW       0        0 RW 0066 NR       0        0 S..    .  -   Synchronized
0148 RW       0        0 RW 0067 NR       0        0 S..    .  -   Synchronized
0149 RW       0        0 RW 0068 NR       0        0 S..    .  -   Synchronized
014A RW       0        0 RW 0069 NR       0        0 S..    .  -   Synchronized
014B RW       0        0 RW 006A NR       0        0 S..    .  -   Synchronized
014C RW       0        0 RW 006B NR       0        0 S..    .  -   Synchronized
014D RW       0        0 RW 006C NR       0        0 S..    .  -   Synchronized
014E RW       0        0 RW 006D NR       0        0 S..    .  -   Synchronized
014F RW       0        0 RW 006E NR       0        0 S..    .  -   Synchronized

Total    ------- -------             ------- -------
Trks           0    1012                   0       0
MBs          0.0    31.6                 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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The `symrdf establish` command initiates an incremental establish operation on SRDF pairs in the composite group that are not synchronized (that is, the suspended pairs on Symmetrix 3143).

```
symrdf -cg SRDF establish -noprompt

An RDF 'Incremental Establish' operation execution is in progress for
   composite
   group 'SRDF'. Please wait...

    Suspend RDF link(s) for device(s) in (3143,01)..................Done.
    Resume RDF link(s) for device(s) in (3143,01)..................Not Done.
    Merge track tables between source and target in (3143,01).......Started.
    Devices: 00F7-00F8 ............................................ Merged.
    Device: 00FA .................................................. Merged.
    Devices: 00FC-0101 ............................................ Merged.
    Devices: 0102-0107 ............................................ Merged.
    Devices: 0108-010D ............................................ Merged.
    Devices: 010E-010F ............................................ Merged.
    Merge track tables between source and target in (3143,01).......Done.
    Resume RDF link(s) for device(s) in (3143,01)..................Done.

The RDF 'Incremental Establish' operation successfully initiated for
   composite
   group 'SRDF'.
```

- Another `symrdf query` command shows that the previously suspended pairs are now in the process of synchronizing. A period (.) in the “Cons” column indicates that consistency protection is disabled.

```
symrdf -cg SRDF query

Composite Group Name     :  SRDF
Composite Group Type     :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID              : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID       : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number     :  1 (A)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------ ----- ------ ------------
     ST                LI      ST                       C  S
      A                 N       A                       o  u
      T  R1 Inv  R2 Inv K       T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks  S Dev   E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ----------------------- ----- ------ ------------
00F7 RW       0       0 RW 0062 NR       0       0 S..   .  -   Synchronized
00FA RW       0      45 RW 0065 NR       0       0 S..   .  -   SyncInProg
00FC RW       0      46 RW 0067 NR       0       0 S..   .  -   SyncInProg
00FD RW       0      46 RW 0068 NR       0       0 S..   .  -   SyncInProg
00FE RW       0       1 RW 0069 NR       0       0 S..   .  -   SyncInProg
00FF RW       0       1 RW 006A NR       0       0 S..   .  -   SyncInProg
0100 RW       0       1 RW 006B NR       0       0 S..   .  -   SyncInProg
0101 RW       0       1 RW 006C NR       0       0 S..   .  -   SyncInProg
0102 RW       0      46 RW 006D NR       0       0 S..   .  -   SyncInProg
0103 RW       0      46 RW 006E NR       0       0 S..   .  -   SyncInProg
0104 RW       0       1 RW 006F NR       0       0 S..   .  -   SyncInProg
0105 RW       0      46 RW 0070 NR       0       0 S..   .  -   SyncInProg
0106 RW       0       1 RW 0071 NR       0       0 S..   .  -   SyncInProg
0107 RW       0       1 RW 0072 NR       0       0 S..   .  -   SyncInProg
0108 RW       0      46 RW 0073 NR       0       0 S..   .  -   SyncInProg
0109 RW       0       1 RW 0074 NR       0       0 S..   .  -   SyncInProg
010A RW       0       1 RW 0075 NR       0       0 S..   .  -   SyncInProg
010B RW       0      46 RW 0076 NR       0       0 S..   .  -   SyncInProg
```

```
010C RW       0       1 RW 0077 NR       0       0 S..   .  -   SyncInProg
010D RW       0       1 RW 0078 NR       0       0 S..   .  -   SyncInProg
010E RW       0       1 RW 0079 NR       0       0 S..   .  -   SyncInProg
010F RW       0       1 RW 007A NR       0       0 S..   .  -   SyncInProg

Symmetrix ID                 : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID          : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number        :  1 (00)

   Source (R1) View         Target (R2) View       MODES STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI      ST                      C  S
      A                   N       A                      o  u
      T   R1 Inv  R2 Inv  K       T   R1 Inv  R2 Inv     n  s   RDF Pair
Dev   E   Tracks Tracks   S Dev   E   Tracks  Tracks MDA s  p   STATE
----------------------- -- ------------------------ ----- ------ ------------
0137 RW       0       0 RW 0056 NR       0       0 S..   .  -   Synchronized
013A RW       0       0 RW 0059 NR       0       0 S..   .  -   Synchronized
013C RW       0       0 RW 005B NR       0       0 S..   .  -   Synchronized
013D RW       0       0 RW 005C NR       0       0 S..   .  -   Synchronized
013E RW       0       0 RW 005D NR       0       0 S..   .  -   Synchronized
013F RW       0       0 RW 005E NR       0       0 S..   .  -   Synchronized
0140 RW       0       0 RW 005F NR       0       0 S..   .  -   Synchronized
0141 RW       0       0 RW 0060 NR       0       0 S..   .  -   Synchronized
0142 RW       0       0 RW 0061 NR       0       0 S..   .  -   Synchronized
0143 RW       0       0 RW 0062 NR       0       0 S..   .  -   Synchronized
0144 RW       0       0 RW 0063 NR       0       0 S..   .  -   Synchronized
0145 RW       0       0 RW 0064 NR       0       0 S..   .  -   Synchronized
0146 RW       0       0 RW 0065 NR       0       0 S..   .  -   Synchronized
0147 RW       0       0 RW 0066 NR       0       0 S..   .  -   Synchronized
0148 RW       0       0 RW 0067 NR       0       0 S..   .  -   Synchronized
0149 RW       0       0 RW 0068 NR       0       0 S..   .  -   Synchronized
014A RW       0       0 RW 0069 NR       0       0 S..   .  -   Synchronized
014B RW       0       0 RW 006A NR       0       0 S..   .  -   Synchronized
014C RW       0       0 RW 006B NR       0       0 S..   .  -   Synchronized
014D RW       0       0 RW 006C NR       0       0 S..   .  -   Synchronized
014E RW       0       0 RW 006D NR       0       0 S..   .  -   Synchronized
014F RW       0       0 RW 006E NR       0       0 S..   .  -   Synchronized

Total     ------- -------            ------- -------
Trks            0     380                  0       0
MBs           0.0    11.9                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

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

- The symcg show command confirms that the consistency state of the devices is currently Disabled.

symcg show SRDF

```
Composite Group Name:  SRDF

  Composite Group Type                                      : RDF1
  Valid                                                     : Yes
  CG in PowerPath                                           : Yes
  CG in GNS                                                 : No

  Number of RDF (RA) Groups                                 :    2
  Number of STD Devices                                     :   44
  Number of BCV's (Locally-associated)                      :    0
  Number of VDEV's (Locally-associated)                     :    0
  Number of RBCV's (Remotely-associated STD-RDF)            :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)           :    0
  Number of RRBCV's (Remotely-associated RBCV)              :    0

  Number of Symmetrix Units (2):
     {

     1) Symmetrix ID                                          : 000000003143
        Microcode Version                                     : 5267
        Number of STD Devices                                 :   22
        Number of BCV's (Locally-associated)                  :    0
        Number of VDEV's (Locally-associated)                 :    0
        Number of RBCV's (Remotely-associated STD_RDF) :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)    :    0

        Number of RDF (RA) Groups (1):
           {

           1) RDF (RA) Group Number  :  1            (A)
              Remote Symmetrix ID    : 000000003156
              Microcode Version      : 5267

              STD Devices (22):
                 {
               -------------------------------------------------------------
                                       Sym  Device        Consistency   Cap
                PdevName               Dev  Config        State         (MB)
               -------------------------------------------------------------
                /dev/vx/rdmp/c15t1d24s2 00F7 RDF1          Disabled     12946
                /dev/vx/rdmp/c15t1d25s2 00FA RDF1          Disabled      8631
                /dev/vx/rdmp/c15t1d26s2 00FC RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d27s2 00FD RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d28s2 00FE RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d29s2 00FF RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d30s2 0100 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d31s2 0101 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d32s2 0102 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d33s2 0103 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d34s2 0104 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d35s2 0105 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d36s2 0106 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d37s2 0107 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d38s2 0108 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d39s2 0109 RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d40s2 010A RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d41s2 010B RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d42s2 010C RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d43s2 010D RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d44s2 010E RDF1          Disabled      4315
                /dev/vx/rdmp/c15t1d45s2 010F RDF1          Disabled      4315
```

```
                  }
              }

        2) Symmetrix ID                                       : 000187900035
           Microcode Version                                  : 5670
           Number of STD Devices                              :   22
           Number of BCV's (Locally-associated)               :    0
           Number of VDEV's (Locally-associated)              :    0
           Number of RBCV's (Remotely-associated STD_RDF)     :    0
           Number of BRBCV's (Remotely-associated BCV-RDF):        0
           Number of RRBCV's (Remotely-associated RBCV)       :    0

           Number of RDF (RA) Groups (1):
              {

              1) RDF (RA) Group Number   :  1           (00)
                 Remote Symmetrix ID     : 000187900041
                 Microcode Version       : 5670

                 STD Devices (22):
                    {
                  -------------------------------------------------------------
                                           Sym  Device        Consistency   Cap
                   PdevName                Dev  Config        State         (MB)
                  -------------------------------------------------------------
                  /dev/vx/rdmp/c15t2d24s2 0137 RDF1+Mir      Disabled     12946
                  /dev/vx/rdmp/c15t2d25s2 013A RDF1+Mir      Disabled      8631
                  /dev/vx/rdmp/c15t2d26s2 013C RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d27s2 013D RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d28s2 013E RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d29s2 013F RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d30s2 0140 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d31s2 0141 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d32s2 0142 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d33s2 0143 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d34s2 0144 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d35s2 0145 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d36s2 0146 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d37s2 0147 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d38s2 0148 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d39s2 0149 RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d40s2 014A RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d41s2 014B RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d42s2 014C RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d43s2 014D RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d44s2 014E RDF1+Mir      Disabled      4315
                  /dev/vx/rdmp/c15t2d45s2 014F RDF1+Mir      Disabled      4315
                    }
              }
        }
```

- The `symcg enable` command enables consistency protection for device pairs in the composite group.

`symcg -cg SRDF enable -noprompt`

```
A consistency 'Enable' operation execution is in progress for composite group
   'SRDF'. Please wait...

The consistency 'Enable' operation successfully executed for composite group
   'SRDF'.
```

- Another `symrdf query` command displays all pairs in the Synchronized state. As indicated in the Legend, an X in the "Cons" column indicates that all pairs are now enabled for consistency protection.

symcg -cg SRDF query

```
Composite Group Name     :  SRDF
Composite Group Type     :  RDF1
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID                : 000000003143    (Microcode Version: 5267)
Remote Symmetrix ID         : 000000003156    (Microcode Version: 5267)
RDF (RA) Group Number       :  1 (A)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI       ST                       C  S
      A                   N        A                       o  u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ---------------------- ----- ------ ------------
00F7 RW       0       0 RW 0062 NR       0       0 S..   X  -   Synchronized
00FA RW       0       0 RW 0065 NR       0       0 S..   X  -   Synchronized
00FC RW       0       0 RW 0067 NR       0       0 S..   X  -   Synchronized
00FD RW       0       0 RW 0068 NR       0       0 S..   X  -   Synchronized
00FE RW       0       0 RW 0069 NR       0       0 S..   X  -   Synchronized
00FF RW       0       0 RW 006A NR       0       0 S..   X  -   Synchronized
0100 RW       0       0 RW 006B NR       0       0 S..   X  -   Synchronized
0101 RW       0       0 RW 006C NR       0       0 S..   X  -   Synchronized
0102 RW       0       0 RW 006D NR       0       0 S..   X  -   Synchronized
0103 RW       0       0 RW 006E NR       0       0 S..   X  -   Synchronized
0104 RW       0       0 RW 006F NR       0       0 S..   X  -   Synchronized
0105 RW       0       0 RW 0070 NR       0       0 S..   X  -   Synchronized
0106 RW       0       0 RW 0071 NR       0       0 S..   X  -   Synchronized
0107 RW       0       0 RW 0072 NR       0       0 S..   X  -   Synchronized
0108 RW       0       0 RW 0073 NR       0       0 S..   X  -   Synchronized
0109 RW       0       0 RW 0074 NR       0       0 S..   X  -   Synchronized
010A RW       0       0 RW 0075 NR       0       0 S..   X  -   Synchronized
010B RW       0       0 RW 0076 NR       0       0 S..   X  -   Synchronized
010C RW       0       0 RW 0077 NR       0       0 S..   X  -   Synchronized
010D RW       0       0 RW 0078 NR       0       0 S..   X  -   Synchronized
010E RW       0       0 RW 0079 NR       0       0 S..   X  -   Synchronized
010F RW       0       0 RW 007A NR       0       0 S..   X  -   Synchronized

Symmetrix ID                : 000187900035    (Microcode Version: 5670)
Remote Symmetrix ID         : 000187900041    (Microcode Version: 5670)
RDF (RA) Group Number       :  1 (00)

   Source (R1) View         Target (R2) View       MODES  STATES
-----------------------  ------------------------- ----- ------ ------------
     ST                  LI       ST                       C  S
      A                   N        A                       o  u
      T  R1 Inv  R2 Inv   K        T  R1 Inv  R2 Inv       n  s   RDF Pair
Dev   E  Tracks Tracks    S Dev    E  Tracks  Tracks MDA   s  p   STATE
----------------------- -- ---------------------- ----- ------ ------------
0137 RW       0       0 RW 0056 NR       0       0 S..   X  -   Synchronized
013A RW       0       0 RW 0059 NR       0       0 S..   X  -   Synchronized
013C RW       0       0 RW 005B NR       0       0 S..   X  -   Synchronized
013D RW       0       0 RW 005C NR       0       0 S..   X  -   Synchronized
013E RW       0       0 RW 005D NR       0       0 S..   X  -   Synchronized
013F RW       0       0 RW 005E NR       0       0 S..   X  -   Synchronized
0140 RW       0       0 RW 005F NR       0       0 S..   X  -   Synchronized
0141 RW       0       0 RW 0060 NR       0       0 S..   X  -   Synchronized
0142 RW       0       0 RW 0061 NR       0       0 S..   X  -   Synchronized
0143 RW       0       0 RW 0062 NR       0       0 S..   X  -   Synchronized
0144 RW       0       0 RW 0063 NR       0       0 S..   X  -   Synchronized
```

```
0145 RW       0       0 RW 0064 NR       0       0 S..    X   -    Synchronized
0146 RW       0       0 RW 0065 NR       0       0 S..    X   -    Synchronized
0147 RW       0       0 RW 0066 NR       0       0 S..    X   -    Synchronized
0148 RW       0       0 RW 0067 NR       0       0 S..    X   -    Synchronized
0149 RW       0       0 RW 0068 NR       0       0 S..    X   -    Synchronized
014A RW       0       0 RW 0069 NR       0       0 S..    X   -    Synchronized
014B RW       0       0 RW 006A NR       0       0 S..    X   -    Synchronized
014C RW       0       0 RW 006B NR       0       0 S..    X   -    Synchronized
014D RW       0       0 RW 006C NR       0       0 S..    X   -    Synchronized
014E RW       0       0 RW 006D NR       0       0 S..    X   -    Synchronized
014F RW       0       0 RW 006E NR       0       0 S..    X   -    Synchronized

Total    ------- -------            ------- -------
Trks           0       0                  0       0
MBs          0.0     0.0                0.0     0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off

Legend for STATES:

 Cons(istency State): X = Enabled, M = Mixed, . = Disabled, - = N/A
 Susp(end State)    : X = Online, . = Offline, P = Offline Pending, - = N/A
```

A

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 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	✓		✓[a]	
Incremental Establish	✓		✓	
Split	✓	✓	✓	✓
Restore	✓[a]		✓	
Incremental Restore	✓		✓	
Failover	✓	✓	✓	✓
Failback	✓		✓	
Suspend	✓	✓	✓	✓
Resume	✓		✓	
Swap	✓		✓	
Create pair	✓		✓	
Delete pair	✓	✓	✓	✓
Update R1 Mirror	✓		✓	
Merge Track Tables	✓		✓	
RW Disable R2 Mirror			✓	✓
RW Enable R1 Mirror	✓	✓		
RW Enable R2 Mirror			✓	✓
Write Disable R1 Mirror	✓	✓		
Write Disable R2 Mirror			✓	✓
Refresh R1 Mirror	✓			
Refresh R2 Mirror			✓	
Invalidate R1 Mirror	✓	✓		
Invalidate R2 Mirror			✓	✓
Ready R1 Mirror	✓	✓		
Ready R2 Mirror			✓	✓
Not Ready R1 Mirror	✓	✓		
Not Ready R2 Mirror			✓	✓

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 `-copy` 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-4 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	✓	✓[a]	✓	
Incremental Establish	✓	✓[a]	✓	
Split	✓	✓	✓	✓
Restore	✓		✓	
Incremental Restore	✓		✓	
Failover	✓	✓	✓	✓
Failback	✓		✓	
Suspend	✓	✓	✓	✓

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
Resume	✓	✓[a]	✓	
Swap	✓		✓	
Create pair	✓		✓	
Delete pair	✓	✓	✓	✓
Update R1 Mirror	✓		✓	
Merge Track Tables	✓		✓	
RW Disable R2 Mirror			✓	✓
RW Enable R1 Mirror	✓	✓		
RW Enable R2 Mirror			✓	✓
Write Disable R1 Mirror	✓	✓		
Write Disable R2 Mirror			✓	✓
Refresh R1 Mirror	✓			
Refresh R2 Mirror			✓	
Invalidate R1 Mirror	✓	✓		
Invalidate R2 Mirror			✓	✓
Ready R1 Mirror	✓	✓		
Ready R2 Mirror			✓	✓
Not Ready R1 Mirror	✓	✓		
Not Ready R2 Mirror			✓	✓

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 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	✓	✓[a]
CreateInProg	✓	✓[a]
NotCreated	N/A	N/A
Copied	✓	✓[a]
CopyInProg	✓	✓[a]
CopyOnAccess	✓	N/A
CopyOnWrite	✓	✓
Restored	✓	✓[a]
RestInProg	✓	✓[a]
Invalid	N/A	N/A
Failed	N/A	N/A
TermInProg	✓	✓[a]

a. Asynchronous mode is not allowed to be set if copy pairs were designated as TimeFinder/Clone CopyOnAccess.

B

SRDF/Star State Reference

This chapter describes the allowable SRDF/Star configuration system states for using the `symstar` command arguments.

SRDF/Star States

To perform a `symstar` command, the SRDF/Star configuration needs to be in an allowable system state. Otherwise, a message is returned, stating that SRDF/Star is not in a state that permits the particular operation that you are attempting to perform. The following sections detail the allowable states for each SRDF/Star control operation. *System State* can be displayed by using the `symstar query` command.

Normal Operations

This section details the allowable states for each action involved in taking an SRDF/Star system into and out of the *STAR Protected* state. It also includes the actions required to isolate a remote site for testing or other required data processing.

Figure B-1 illustrates the allowable states for each SRDF/Star action in this operational context.

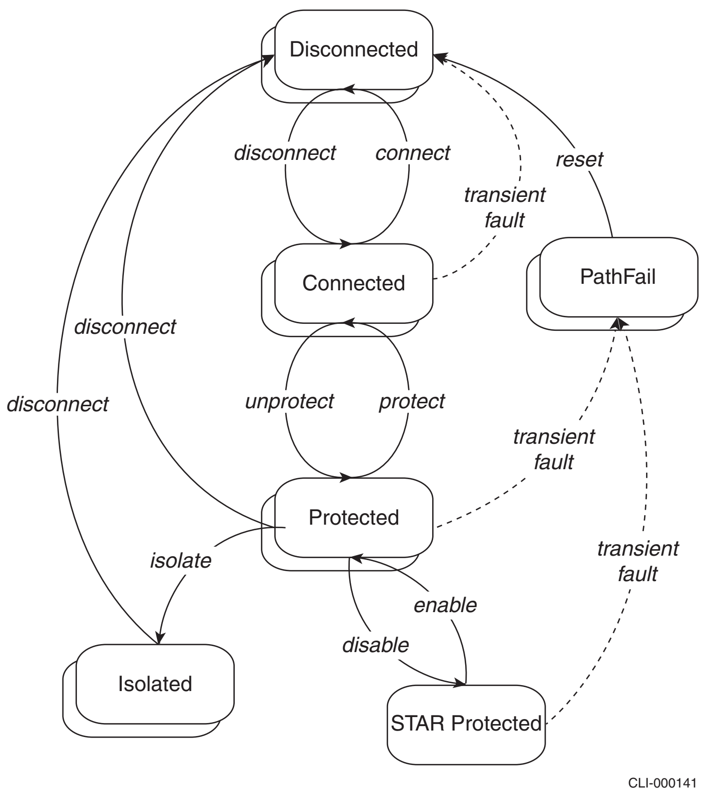

Figure B-1 SRDF/Star Normal Operation Model

Protect

You can use the `symstar protect` command while the target sites are in the following state:

- Both are *Connected*
- One is *Connected*
- One is *Connected* and the other is *PathFail*

You can only use the `protect` action on a site that is Connected. For example, if the current state of the synchronous target site is PathFail, you cannot protect the synchronous target site without first performing a `reset` action.

Unprotect You can use the `symstar unprotect` command when the specified target site is in the *Protected* state. If you are running the workload at the asynchronous target site (Site C), only one path can be protected (in asynchronous mode) at a time. However, you can toggle the protection between Site A and Site B. That is, you can unprotect one and then protect the other.

Enable You can use the `symstar enable` command while both target sites are in the *Protected* state. However, if you are running the workload at the asynchronous target site (Site C), only one path can be protected at a time. Therefore, the `enable` action is blocked when running the workload from the asynchronous target site.

Disable You can use the `symstar disable` command while the *System State* is *STAR_Protected*.

Isolate The site that you are isolating must be in the *Protected* state. You can use the `symstar isolate` command while the remote target site are in the following states:

- Both are *Protected*
- The target is *Protected*, and the other is *Connected*, *Disconnected*, or *Isolated (that is, the state of the other site does not matter)*

Reset You can use the `symstar reset` command while the target sites are in the following states:

- Both are *PathFail*
- One is *PathFail*, and the other is *Protected* or *Isolated*

You should use the `reset` action after correcting the physical cause of a transient fault. Unless the `-force` option is specified, the `reset` action is rejected if the RDF path to the site is in a *Partitioned* state (unless the `-force` option is specified).

Connect The site to which you are connecting must be in the *Disconnected* state. You can use the `symstar connect` command while the target sites are in the following states:

- Both are *Protected*
- Both are *Connected*
- One is *Disconnected*, and the other is *Protected*

Disconnect The site from which you are disconnecting must be in the *Connected* or *Protected* state. You can use the `symstar disconnect` command while the target sites are in the following states:

- Both are *Protected*
- Both are *Connected*
- One is *Protected* or *Connected*, and the other is *Disconnected*

Unplanned WorkLoad Switch Operations

This section details the allowable states for each action involved with responding to an unplanned event that causes you to switch the workload from the current workload site a new workload site.

Figure B-2 illustrates the allowable states for each SRDF/Star action in this operational context.

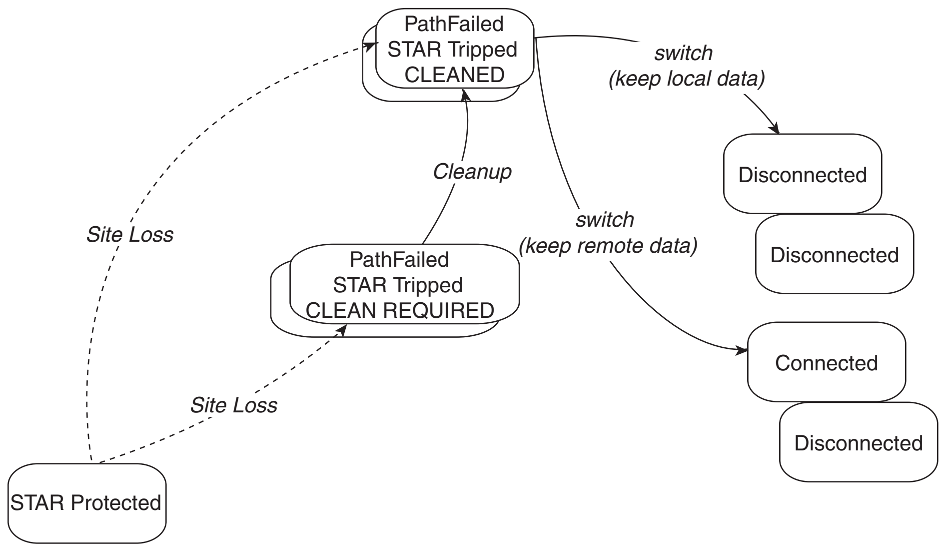

Figure B-2 Unplanned Workload Switch Operations

Cleanup

You can use the `symstar cleanup` command when the state of the synchronous target site is PathFail and the asynchronous target site is *PathFail;CleanReq*.

Unplanned Switch to Sync Target Site; Keep Sync Site Data

You can use a `symstar switch` command that specifies the synchronous target site and keeps the synchronous site data while the *System State* components are 1st_target_site: *PathFail*, 2nd_target_site: *PathFail*, and STAR:*Tripped*.

Unplanned Switch to Sync Target Site; Keep Async Site Data

You can use a `symstar switch` command that specifies the synchronous target site and keeps the asynchronous site data while the *System State* components are *PathFail*, *PathFail*, and *Tripped*. When you keep the remote site data (asynchronous site data in this case), control is not returned until the `switch` action has completely synchronized the data.

Unplanned Switch to Async Target Site; Keep Async Site Data

You can use a `symstar switch` command that specifies the asynchronous target site and keeps the asynchronous site data while the *System State* components are 1st_target_site: *PathFail*, 2nd_target_site: *PathFail*, and STAR:*Tripped*.

Unplanned Switch to Async Target Site; Keep Sync Site Data

You can use a `symstar switch` command that specifies the asynchronous target site and keeps the synchronous site data while the *System State* components are *PathFail*, *PathFail*, and *Tripped*. When you keep the remote site data (synchronous site data in this case), control is not returned until the `switch` action has completely synchronized the data.

Planned Workload Switch Operations

This section details the allowable states for each action involved with switching the workload from the current workload site to a new workload site in a planned procedure.

Figure B-3 illustrates the allowable states for each SRDF/Star action in this operational context.

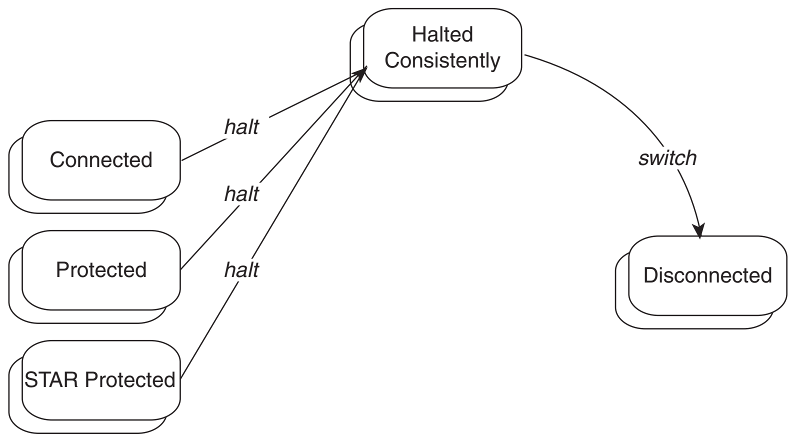

Figure B-3 Planned Workload Switch Operations

Planned Switch

You can use the `symstar switch` command to perform a planned switch operation if both target sites are in the *Halted* state. Any other states relate to an unplanned switch operation.

Halt

To halt the SRDF/Star system, the target sites must either be *Connected* or *Protected*. You can use the `symstar halt` command while the target sites are in the following states:

- Both are *Protected*, and STAR is *Protected*
- Both are *Protected*
- Both are *Connected*
- One is *Protected* and the other is *Connected*

Index

T

U