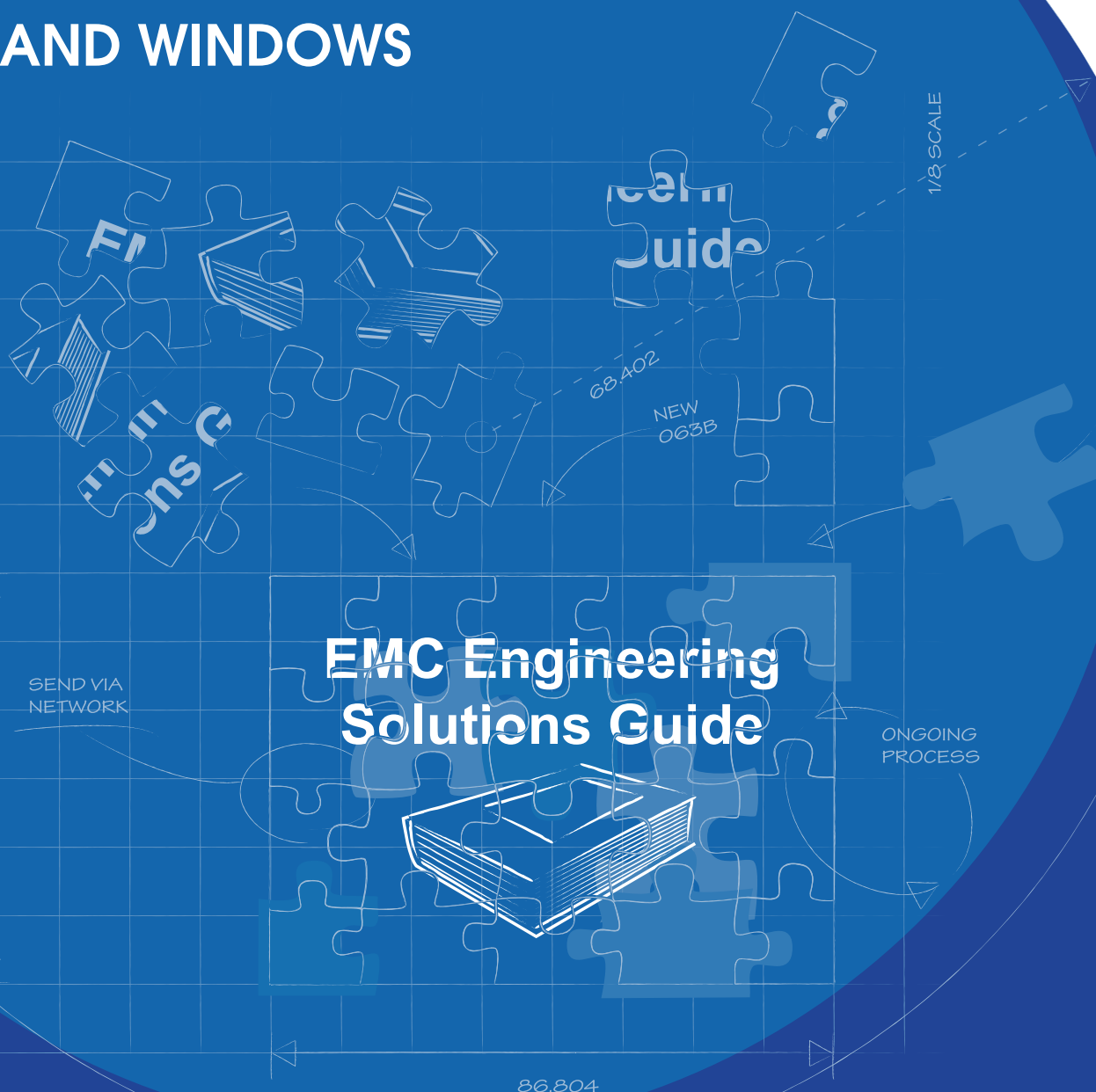


SYMMETRIX AUTOMATED REPLICATION UNIX AND WINDOWS

Version 1.0



Authors

Corsino Climacosa, Flavio Fomin,
Nancy Gessler, Dawn Johnsey,
Kurt Schmidt, and Stuart Shellswell

Corporate Information

Corporate Headquarters:
Hopkinton, MA 01748-9103
(800) 424-EMC2
<http://www.EMC.com>

Copyright © 2003 EMC Corporation. All rights reserved.

Printed December, 2003

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

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

Preface	vii
----------------------	-----

Chapter 1 Background and Introduction

Overview	1-2
Essential Terms and Concepts.....	1-3
Asynchronous Remote Replication Solutions	1-4
Symmetrix Automated Replication	1-4
Role of Automation in Symmetrix Automated Replication Installations	1-4
Disaster Restart versus Disaster Recovery	1-4
Lower Communications Bandwidth Requirements	1-5
Single Hop Configuration	1-5
Multihop Configuration	1-7

Chapter 2 EMC Products to Support Symmetrix Automated Replications for UNIX and Windows

EMC Symmetrix System	2-2
EMC Solutions Enabler	2-3
EMC TimeFinder	2-3
EMC Symmetrix Remote Data Facility	2-5
EMC Change Tracker	2-7
PowerPath.....	2-8

Chapter 3 Planning a Symmetrix Automated Replication Installation

Overview	3-2
Understanding the Current Environment.....	3-3

Existing Hardware Configuration	3-3
Change Tracker Data Collection	3-4
Planning the Network Configuration.....	3-5
Channel Extenders	3-5
Circuit Distance	3-6
FarPoint Buffer.....	3-6
Compression Rate.....	3-6
Disk Emulation	3-6
Understanding the Characteristics of a Symmetrix Automated	
Replication Workload.....	3-7
Basic Components	3-7
TimeFinder and SRDF Commands.....	3-8
Copy Tasks	3-8
Invalid Tracks.....	3-8
Performance Modeling	3-9
Applying the Model.....	3-9
Considerations for Performance Modeling	3-11

Chapter 4 TimeFinder Consistent Split

Overview.....	4-2
Practical Uses.....	4-3
Consistent Split	4-4
Creating a Local DBMS-Restartable BCV Copy	4-7
Creating a DBMS-Restartable BCV Copy on a	
Remote Symmetrix	4-8
Creating DBMS-Restartable Copies on Both Sides	4-9
Using Consistent Split When Performing SRDF/AR.....	4-10
Performing a Split with Database Assist.....	4-11
Freezing an Oracle Database	4-12
Freezing an Informix Database.....	4-12
Freezing an SQL 2000 Database	4-12
Freezing an IBM UDB Database.....	4-13
Freezing a Sybase Database	4-13

Chapter 5 Using SYMCLI to Perform Symmetrix Automated Replication

Overview.....	5-2
Establishing a BCV Pair with an RDF BCV	5-3
Copying Data to a Remotely-Associated BCV	5-4
Copying Data in a Complex Multihop Environment.....	5-6
Targeting Remote Devices when the Data Source Is the Local	

Standard RDF Device	5-6
Targeting Remote Devices when Remote Copying Is from a Local R1 BCV	5-7
Using Symmetrix Automated Replication	5-10
Single-Hop Configuration	5-10
Multihop Configuration.....	5-11
Initial Setup Tasks for Using Symmetrix Automated Replication	5-12
Accessing a Concurrent BCV while Running SRDF/AR.....	5-19

Chapter 6 Symmetrix Automated Replication Restart Procedures

Overview	6-2
Prior to the Disaster	6-3
Corresponding Device Groups or Device Files on Target Host	6-3
Application Testing Procedures versus Actual Disaster Restart Command Procedures	6-3
Application Testing Procedures	6-4
Actual Disaster Restart Command Procedures	6-4
Hardware and Software Configuration at the Target Site.	6-4
Disaster Restart Procedures for a Single-Hop Configuration...	6-5
Determining the Most Current Consistent Copy and Restarting Applications.....	6-5
Make the Devices Ready to the Host	6-12
Restarting the Applications	6-12
Disaster Restart Off the BCVs	6-12
Disaster Restart Procedures for a Multihop Configuration	6-13
Determining the Most Current Consistent Copy and Restarting Applications.....	6-13
Make the Devices Ready to the Host	6-19
Restarting the Applications	6-19
Disaster Restart Off the BCVs	6-19

Chapter 7 Symmetrix Automated Replication Return Home Procedures

Overview	7-2
Return Home Overview	7-3
SRDF R1 Update/Resynchronization	7-4
Point of Command Execution	7-4
R1 Update Single-Hop Configuration	7-5
R1 Update Multihop Configuration.....	7-7

SRDF R1-R2 Swap.....	7-10
R1-R2 Swap Requirements.....	7-10
R1-R2 Swap Command Format	7-10
R1-R2 Single-Hop Configuration	7-11
R1-R2 Multihop Configuration	7-13
Dynamic RDF	7-16
Dynamic RDF Requirements	7-16
Verifying Dynamic RDF Devices	7-16
Creating a Device File	7-16
Dynamic RDF Command Format	7-17
Dynamic RDF Single-Hop Configuration.....	7-18
Dynamic RDF Multihop Configuration	7-24

Appendix A Command Examples Related to Chapter 4

Example 1: Creating an RDBMS-Restartable Copy on a Local Symmetrix.....	A-2
Example 2: Creating an RDBMS-Restartable BCV Copy on a Remote Symmetrix Using PowerPath Support.....	A-9
Example 3: Performing Splits with Database Assist.....	A-14
Splitting BCV Pairs Using SQL 2000 Database Assist.....	A-14
Splitting BCV Pairs Using Oracle Database Assist.....	A-15

Appendix B Command Examples Related to Chapter 5

Example 1: Single-Hop Configuration	B-2
Example 2: Multihop Configuration with BCVs at Hop 2.....	B-14
Example 3: MultiHop Configuration without BCVs at Hop 2	B-22
Example 4: Accessing Concurrent Non-SRDF / AR BCVs while Running SRDF / AR (Single-Hop Configuration).....	B-31
Example 5: Accessing Concurrent Non-SRDF / AR BCVs while Running SRDF / AR (Multihop Configuration).....	B-35
Example 6: Restarting a Replicate Session when Devices Are Locked	B-40

This solutions guide describes how to plan, implement, and customize Symmetrix® Automated Replication (SRDF®/AR) for UNIX and Windows using the EMC Symmetrix systems EMC Solutions Enabler, TimeFinder™ component, SRDF component, and PowerPath® or Enginuity™ Consistency Assist. The intended audience is storage management personnel, capacity planning personnel, database administrators, and disaster recovery/restart personnel responsible for implementing Symmetrix Automated Replication on UNIX and/or Windows via a command line interface. The information in this document is based on Solutions Enabler V5.3 and PowerPath V3.0.0. This document assumes the UNIX and Windows environments can use EMC PowerPath software or Enginuity Consistency Assist. Typically, this will encompass HP-UX, SUN, AIX, Tru64 and Windows.

This solutions guide provides a general description of Symmetrix Automated Replication and the EMC products that can be used for Symmetrix Automated Replication administration on UNIX and Windows. Planning for EMC Symmetrix Automated Replication is also discussed. The implementation, initialization, and maintenance procedures are outlined, and procedures for restarting and going home in the event of a disaster are also covered.

This solutions guide is divided into the following chapters. Chapter topics range from general EMC product descriptions in chapters 1 and 2 to detailed discussions of implementation procedures in chapters 3 through and 5. These chapters include detailed examples and sample SYMCLI commands used to implement the examples.

Chapter 1, *Background and Introduction*, provides a high-level overview of Symmetrix Automated Replication.

Chapter 2, *EMC Products to Support Symmetrix Automated Replications for UNIX and Windows*, describes EMC products used to support Symmetrix Automated Replication.

Chapter 3, *Planning a Symmetrix Automated Replication Installation*, provides a practical approach to the issue of capacity planning and performance modeling applicable to a Symmetrix Automated Replication solution.

Chapter 4, *TimeFinder Consistent Split*, describes TimeFinder functionality included in versions of EMC Solutions Enabler up to version 5.3 running on Symmetrix units using Enginuity Version 5x66, 5x67, 5568, and 5669. This chapter describes the TimeFinder consistent split operation for obtaining a local or remote DBMS-restartable BCV copy of the database.

Chapter 5, *Using SYMCLI to Perform Symmetrix Automated Replication*, describes SRDF and TimeFinder functionality included in versions of EMC Solutions Enabler up through version 5.3 running on Symmetrix units using Enginuity versions 5x66, 5x67, 5568, and 5669.

Chapter 6, *Symmetrix Automated Replication Restart Procedures*, describes the implementation and initialization of symreplicate processing in single-hop or multihop mode.

Chapter 7, *Symmetrix Automated Replication Return Home Procedures*, describes the required steps to determine the most recent copy of data and the procedures required to restart the environment after a disaster.

Appendix A, *Command Examples Related to Chapter 4*, contains the examples for Chapter 4.

Appendix B, *Command Examples Related to Chapter 5*, contains the examples for Chapter 5.

The appendix provides sample code, which supplements some of the procedures.

References

The information in this document was compiled from the following sources:

- ◆ *Using SYMCLI to Set Up TimeFinder BCV Pairs* (P/N 300-000-072)
- ◆ *Using SYMCLI to Set Up TimeFinder Multiple BCVs and Concurrent BCVs* (P/N 300-000-073)
- ◆ *Using SYMCLI to Perform TimeFinder Control Operations* (P/N 300-000-074)
- ◆ *Using SYMCLI to Perform TimeFinder Consistent Splits* (P/N 300-000-283)
- ◆ *Using SYMCLI to Perform SRDF Control Operations* (P/N 300-000-076)
- ◆ *Using SYMCLI to Implement SRDF Consistency Groups*

This chapter reviews the following topics:

- ◆ 1.1 Overview1-2
- ◆ 1.2 Essential Terms and Concepts1-3
- ◆ 1.3 Asynchronous Remote Replication Solutions1-4

1.1 Overview

Using TimeFinder and Symmetrix Remote Data Facility (SRDF) software to mirror data from a source Symmetrix system to a remotely associated business continuance volume (BCV) located across the hall or across the globe creates a “global” BCV concept. That is, production data can be available on remotely associated BCVs for data mining, e-business content delivery, application development, testing, backups, and numerous other uses. Moreover, an SRDF multihop configuration (multihop mirroring to a third site) provides a restart solution for component or site failures between remotely mirrored devices.

SRDF data propagation allows you to mirror only the data that has changed since the last update to a Symmetrix system in another location. By copying only the changed tracks, you consume less bandwidth and enhance performance. You can perform Symmetrix-to-Symmetrix transmissions synchronously in the local or campus area, or use delayed synchronization modes for long-distance transmission. Multihop mirroring to a third site can take place during off-peak times or over lower-cost transmission lines or via IP-based virtual private networks.

EMC Symmetrix Automated Replication (SRDF/AR) is a process that can be used for remote mirroring to be automatic according to your own predefined copy cycle (every hour, for example). You can use a single hop disaster restart configuration that permits controlled data loss (depending on the cycle time). If greater protection is required, a multihop restart configuration can provide synchronous disaster restart with zero data loss (that is, all completed transactions on the source are completed on the target). Compared to traditional disaster recovery solutions with their long restart time and high data loss, disaster restart solutions using Symmetrix Automated Replication provides remote restart with a short restart time and low data loss. Using a `symreplicate` session to copy data to remote sites provides not only the automation benefits for restart solutions but also the ability to control the frequency and duration of each copy cycle as well as other copy cycle options.

EMC Symmetrix Automated Replication is a process that offers customers data protection with dependent write consistency across a distance. This protection is accomplished by using geographically separated replicas with hardware and software products from EMC Corporation. This document addresses the planning and implementation aspects of Symmetrix Automated Replication and how it engages the capabilities of the Symmetrix storage system, PowerPath, SRDF, and TimeFinder as delivered by EMC Solutions Enabler. Although the Symmetrix Automated Replication process can be implemented via Symmetrix Data Mobility Manager (SDMM) in an open systems environment, or with ResourcePak® Base in a mainframe OS/390 or z/OS operating system, the scope of this document is limited to UNIX and Windows environments using PowerPath or Enginuity Consistency Assist and Solutions Enabler command line interface installations.

1.2 Essential Terms and Concepts

The following terms are used throughout this document:

Dependent Write-Consistent — A data state where data integrity is guaranteed by dependent write I/Os.

Dependent Write I/O — An I/O that cannot be issued until an associated predecessor I/O has completed. Most applications, and in particular database management systems (DBMSs), have imbedded dependent write logic to ensure data integrity in the event a failure occurs in the host or server processor, software, storage subsystem, or if an environmental power failure occurs.

Disaster recovery — The process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known *point of consistency*.

Disaster restart — The restarting of *dependent write-consistent* copies of data and applications, using the automated application of DBMS recovery logs during DBMS initialization to bring the data and application to a *transactional point of consistency*.

Operational complexity — The processes, procedures, and expertise required to perform all necessary recovery and restart functions. The operational complexity must be taken into account in order to select a disaster restart or disaster recovery solution.

Point of Consistency — A point in time to which data can be restored and recovered or restarted and maintain integrity for all data and applications.

Recovery Point Objective (RPO) — The amount of acceptable data loss after the recovery or restart processes are complete. This is the difference between the point of the disaster and the time of the last known *point of consistency*.

Recovery Time Objective (RTO) — The amount of time allowed for performing a restore and recovery or restart to a specified *point of consistency*.

Restore — A process that reinstates a prior copy of the data.

Rolling Disaster — A series of events that lead up to a complete disaster. For example, the loss of a communication link occurs prior to a site failure.

Transactional Consistency — A DBMS state where all transactions are committed or rolled back.

1.3 Asynchronous Remote Replication Solutions

EMC supports synchronous and asynchronous restart solutions. Asynchronous remote replication solutions are distinguished from synchronous remote replication solutions at the application or database level. With synchronous replication, the application waits for an acknowledgment from the storage unit that the data write has completed on the target. Application logic does not allow any dependent write activity until this acknowledgment has been received. If the time to receive this notice is elongated, then response times for users or applications can increase.

The alternative is to use an asynchronous remote replication solution that provides notice to the application that the write has been completed, regardless of whether the remote physical write activity has completed. The application can then allow dependent write activity to proceed immediately.

It is possible to lose consistency and/or data in an asynchronous remote replication solution, so various techniques have been developed in the industry to prevent data loss and/or ensure data consistency. EMC uses SRDF Consistency Groups, TimeFinder Consistent Split, and Symmetrix Automated Replication individually or in combination to address these issues. Refer to Chapter 2 for further information on EMC products.

1.3.1 Symmetrix Automated Replication

Symmetrix Automated Replication (SRDF/AR) automates the movement of a dependent write-consistent data copy across SRDF links in an adaptive copy mode. This creates a restartable image of the data at a remote site in the event of a disaster at the production site. The consistency technology used in Symmetrix Automated Replication ensures that a dependent write-consistent copy and a restartable image of the data is perpetually propagated to the remote site in an automated fashion that is transparent to the host application or database.

The adaptive copy method of transport allows the production and remote site to be great distances apart. The result is a continuously refreshed “data save point” that can be used as the basis for restarting host applications at the remote end of the link. We will refer to the remote site as the restart site. Symmetrix Automated Replication can be used in either single hop or multihop configurations. This will create either a controlled-data-loss or a zero-data-loss environment at the restart site.

1.3.2 Role of Automation in Symmetrix Automated Replication Installations

The purpose of Symmetrix Automated Replication is to provide consistent, restartable images of data at a remote location. Automation ensures that all necessary operations are performed in the proper sequence. Additionally, automation optimizes the steps of the Symmetrix Automated Replication cycle. For example, the TimeFinder establish processes will occur simultaneously at both the production and the restart site. The result is while the final data save point is being created; the next dependent write-consistent copy is being established, thus reducing overall cycle times. Automation allows the next cycle to begin without undue loss of time.

1.3.3 Disaster Restart versus Disaster Recovery

Disaster recovery implies the use of backup technology in which data is copied to tape, and then shipped offsite. When a disaster is declared, the remote site copies are restored and logs are applied to bring the data to a point of consistency. Once all recoveries are completed, the data is validated to ensure it is correct. Coordinating to a common business

point of recovery across all applications and other platforms can be difficult, if not impossible, using traditional recovery methods.

Disaster restart solutions allow the restart of all participating DBMSs to a common point of consistency utilizing the automated application of DBMS recovery logs during DBMS initialization. The restart time is comparable to the length of time required for the application to restart after a power failure.

Restartable images that are dependent write-consistent, which can be used for restart, can be created locally, and then transported to an offsite storage site via tape or SRDF. These restartable images can also be created remotely. Dependent write consistency is ensured by EMC technology and transactional consistency is ensured by the DBMS at restart, similar to recovery from a local power failure.

Transactional consistency is made possible by the dependent write I/O philosophies inherent in logging DBMS systems. Dependent-write I/O is the methodology all logging DBMS systems use to maintain integrity internally. Data writes are dependent on a successful log write in these systems, and therefore, restartability is guaranteed. Restarting a database environment instead of recovering it enhances availability and reduces operational complexity. Restart is significantly faster than traditional recovery methods.

1.3.4 Lower Communications Bandwidth Requirements

The Symmetrix Automation Replication link is by definition adaptive copy, thus, slower or fewer links can be used. The amount of data loss depends on the size of the communication links between the source and target Symmetrix units. Less bandwidth means longer cycles and more potential data loss, whereas more bandwidth means shorter cycles and less potential data loss. Symmetrix Automated Replication allows users the ability to size their bandwidth based upon their own installation needs and recovery point objectives.

TimeFinder serves the function of creating dependent write-consistent point-in-time copies for the next cycle. Once the first set of data is moved during the initialization of Symmetrix Automated Replication, all subsequent updates are based upon changed tracks only. Only if a track has been changed will it be moved to the restart site. Less data needs to be transferred in the Symmetrix Automated Replication solution compared to log-based systems that must order and then transmit every I/O whether it involves the same physical track or not.

1.3.5 Single Hop Configuration

Symmetrix Automated Replication single hop is a true asynchronous solution providing protection and restartability with some degree of controlled data loss (as would occur in any completely asynchronous remote solution). Once the dependent write-consistent copy is created on the production site, it is sent in adaptive copy mode to the restart site. The existing dependent write-consistent copy is saved to a set of TimeFinder BCVs on the restart site while the next dependent write-consistent copy is being updated on the production site.

The only data that could be lost is that which is en route from the production site to the restart site. Such data loss is controlled in that the previous Symmetrix Automated Replication cycle would have preserved a dependent write-consistent copy as of the completion of the last cycle. The amount of data loss corresponds directly to the cycle time. The cycle time is a function of the bandwidth of the SRDF links, the distance of the replication, the amount of changed data, and the locality of reference of the changed data. Figure 1-1 depicts a single hop configuration and is described in the steps following the figure.

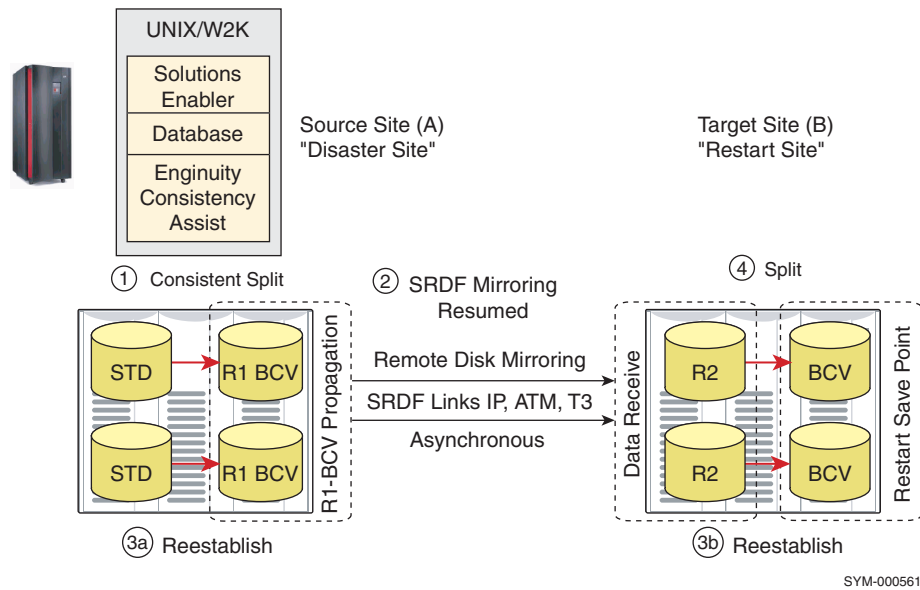


Figure 1-1 Symmetrix Automated Replication Cycle

1. Create a restartable copy. The source site R1-BCVs are consistent split from the source standard devices. A consistent split will defer only the write I/Os during the instant split utilizing Enginuity Consistency Assist (ECA). If a write I/O is received per logical device, subsequent read I/O will also be deferred for that logical volume. I/O is released after the instant split and the background split continues.
2. Transmit the restartable copy. The target R2 devices are synchronized with the source R1-BCVs.
3. Create the next restartable copy at the source and save the restartable copy at the target. The reestablish processes on the source and target site occur in parallel to shorten the length of the cycle.
 - a. The source R1-BCVs are reestablished with their standard devices to capture the next dependent write-consistent copy.
 - b. The target BCVs are re-established with the target R2 devices to create a dependent write-consistent save copy.
4. Secure the save restartable copy at the target. The target BCVs are split from the target R2 devices.

This cycle continues based on parameters provided.

1.3.6 Multihop Configuration

Symmetrix Automated Replication multihop creates the ability to provide protection at great distances without any data loss. A middle or bunker site transitions the synchronous replication solution to an adaptive copy replication solution. Every I/O that occurs at the production site will occur in synchronous mode to the bunker site. SRDF Consistency Groups are required for implementations spanning multiple Symmetrix systems to provide protection from a rolling disaster.

The bunker site is used to create the dependent write-consistent copy and propagate it to the restart site in adaptive copy mode. Provided the disaster only involves the production site, every dependent write transaction that occurred at the production site will be propagated to the restart site with one final transfer from the bunker site to the restart site. The restart of the environment will then transition the dependent write-consistent copy to transactional consistency.

EMC provides many hardware and software products that support Symmetrix Automated Replication environments. These products are described below.

- ◆ 2.1 EMC Symmetrix System2-2
- ◆ 2.2 EMC Solutions Enabler2-3
- ◆ 2.3 PowerPath.....2-8

2.1 EMC Symmetrix System

The EMC Symmetrix system, with its business continuance features and advanced software capability, has lead the way in providing real solutions to the difficult problem of protecting mission-critical data. Symmetrix business continuance features improve a wide range of processes, from scheduled backup to disaster recovery and restart, while providing high-performance access to data.

The Symmetrix system uses RAID 1 mirroring technology to provide high levels of data availability. In a RAID 1 protection scheme, identical data is stored on a redundant mirror. Should either mirror fail; the other mirror provides continuous availability transparent to the host application. RAID 1 also improves performance by allowing either mirror to service read requests, reducing contention at the physical device level.

The Symmetrix system features a *cache all* architecture. All data flows through cache and all writes are *fast writes*. There is no concept of a *write miss*. The cache management algorithms manage the cache effectively through techniques such as sequential prefetch and multiple LRU structures ensuring very high read-hit ratios. All data is cached to improve performance.

2.2 EMC Solutions Enabler

The Solutions Enabler (known as SYMCLI) is a specialized library or set of UNIX-formatted commands that can be invoked one at a time. The SYMCLI is used in single command line entries and scripts to monitor and perform control operations on devices and data objects in order to manage the storage complex. It also monitors device configuration and status of devices that make up the storage environment.

SYMCLI resides on a host system to monitor and perform control operations on Symmetrix units. SYMCLI commands are invoked from the host operating system command line (shell). The commands are linked with SYMAPI library functions. The library functions use system calls that generate low-level I/O SCSI commands to the Symmetrix units.

To reduce the number of queries from the host to the Symmetrix units, configuration and status information is maintained in a Symmetrix host database file.

Table 2-1 lists the SYMCLI Base commands that are discussed in this document.

Table 2-1 SYMCLI Base Commands

Command	Argument	Description
symdg	create delete rename release list show	Performs operations on a device group (dg): - Creates an empty device group. - Deletes a device group. - Renames a device group. - Releases a device external lock associated with all devices in a device group. - Displays a list of all device groups known to this host. - Displays detailed information about a device group and any gatekeeper or BCV devices associated with the device group.
symlld	add list remove rename show	Performs operations on a device in a device group (dg): - Adds devices to a device group and assigns the device a logical name. - Lists all devices in a device group and any associated BCV devices. - Removes a device from a device group. - Renames a device in the device group. - Displays detailed information about a device in a device group.
symbcv	list associate disassociate associate -rdf disassociate -rdf	Performs support operations on BCV pairs: - Lists BCV devices. - Associates BCV devices to a device group—required in order to perform operations on the BCV device. - Disassociates BCV devices from a device group. - Associates remotely attached BCV devices to an RDF device group. - Disassociates remotely attached BCV devices from an RDF device group.

2.2.1 EMC TimeFinder

The SYMCLI TimeFinder component extends the basic SYMCLI command set to include the `symmir` command, which allows you to perform control operations on BCVs within your TimeFinder environment. Symmetrix TimeFinder is a business continuance solution that allows the use of BCV Symmetrix devices. A BCV device contains a copy of data from a standard Symmetrix device that is online for regular I/O operation from its host(s). Uses for the BCV copies can include backup, restore, decision support, and applications testing. Each BCV device has its own host address, and is configured as a stand-alone Symmetrix device.

A business continuance sequence first involves establishing the BCV device as a mirror of a specific standard Symmetrix device. As a result, the BCV device becomes inaccessible via its original device address while it is in an established pair. Once the BCV device is synchronized, it may be separated (*split*) later from the standard device with which it is paired. The BCV device with the synchronized data becomes available for backup or other host processes through its original device address.

Once host processing on the BCV device is complete, the BCV may again be mirrored to a standard Symmetrix device, either the same device to which it was previously paired or with a different device.

Table 2-2 lists the SYMCLI TimeFinder commands that are discussed in this document.

Table 2-2 SYMCLI TimeFinder Commands

Command	Argument	Description
symmir	establish [-rdf] [-full] [-bcv] restore [-rdf] [-full] [-bcv] split [-rdf] [-bcv] query	Performs control operations on BCV device pairs including: - Establishes (mirror) one or all standard devices in a device group with one or more BCV devices that are associated locally or remotely with the group. The operation can be a full or incremental establish. - Restores one or all standard devices in a device group from one or more BCV devices that are associated locally or remotely with the group. The operation can be a full or incremental restore. - Splits one or all BCV devices from the standard device(s) in a device group. - Returns information about the state of mirroring of one or all BCV device pairs in a device group.

2.2.1.1 Enterprise TimeFinder Consistent Split

Enterprise TimeFinder consistent split (E-ConSplit) is used to maintain the integrity (restartability) of a database or application on the set of BCV devices. Since Enginuity 5x66, E-ConSplits are only implemented with an instant (*-instant*) split where you must also specify either a database or PowerPath device(s) and any pre-action and post-action scripts:

```
symmir -g DgName split -instant
      -rdb -dbtype DbType[-db DbName]
      [-preaction Script] [-postaction Script]
```

To target the PowerPath standard devices of the group, or just specific PowerPath device name(s) as a target, use the following syntax:

```
symmir -g DgName split -instant
      -ppath STDDEVS|<PowerPathPdevName...>
      [-preaction Script] [-postaction Script]
```

Consistent split operations require v2.0.1 or higher PowerPath-connected devices on a Symmetrix system with Enginuity 5x66 or higher. Through the assistance of PowerPath, a *symmir consistent split* (supplied with database or PowerPath parameters) initially suspends I/O to the devices that are supplied. This will create a dependent write-consistent copy of data on the BCV devices.

The E-ConSplit command allows the name of scripts to be specified using the *-preaction* and *-postaction* script options. The script commands are executed prior to the freeze and after the thaw operation, respectively.

2.2.2 EMC Symmetrix Remote Data Facility

SRDF is an extension to the business continuance solution that maintains a mirror image of data at the device level in Symmetrix systems that may be located in physically separate sites.

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

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

Table 2-3 lists the SYMCLI SRDF commands that are discussed in this document.

Table 2-3 SYMCLI SRDF Commands

Command	Argument	Description
symrdf	establish [-brbcv -rbcv -bcv] [-full]	Performs the following control operations on RDF devices: Establishes (mirror) 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.
	restore [-brbcv -rbcv -bcv] [-full]	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.
	split [-brbcv -rbcv -bcv]	Splits an RDF pair, which stops mirroring for the RDF pair(s) in a device group.
	swap	Swaps the source (R1) and target (R2) destinations between the target and source. Establishes or deletes dynamic SRDF pairs.
symreplicate	[start stop restart] query	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, stop, and restart the replicate session.

2.2.2.1 Enterprise SRDF Consistency Groups

In an SRDF environment using PowerPath-connected devices of the Symmetrix units, enterprise SRDF Consistency Group (E-ConGroup) configurations are possible. Controlled by the host via SRDF links, E-ConGroups operate in unison to maintain the integrity and dependent write consistency of a database or application distributed across multiple SRDF units.

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 data files. 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.

SRDF-established E-ConGroups could prevent this from occurring by the PowerPath pseudo-device driver 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 E-ConGroup before the intercepted I/O and return control to the application.

In this way, E-ConGroups prevent dependent I/O from getting out of sync, thus ensuring the integrity and dependent write consistency of the data at the remote site.

The following steps refer to Figure 2-1 on page 2-7 and describe a series of events that define an E-ConGroup, execute the E-ConGroup during a rolling disaster, and recover from the rolling disaster. Circled numbers in Figure 2-1 map to the steps listed below.

1. An E-ConGroup is defined to the SYMAPI database via the SYMCI `symcg` command. Devices R1(A), R1(C), and R1(E) are defined to a named E-ConGroup. The E-ConGroup is then enabled.
2. The last SRDF link in an RA group breaks. The break could indicate the beginning of a rolling disaster.
3. The first write I/O to discover the break is intercepted by PowerPath pseudo-device driver. This example shows the log write being intercepted.
4. PowerPath immediately, but temporarily, freezes all I/O to all source devices defined in the E-ConGroup. The intercepted I/O does not complete to the application, preventing dependent I/Os from being issued. Freezing I/O in this manner ensures data integrity preservation at the restart site. Applications writing to non-E-ConGroup-defined devices will not cause an E-ConGroup *trip*, and thus will not suspend SRDF relationships.
5. While I/O is temporarily halted to the devices in the E-ConGroup, SYMCLI issues commands through the source Symmetrix system to suspend SRDF relationships for all disk devices defined in the E-ConGroup. The SRDF suspension requires two I/Os per source Symmetrix system to suspend all necessary (source-to-target) mirroring relationships. The length of time I/O is suspended is a function of the number of Symmetrix systems contained in the E-ConGroup.
6. Once SRDF relationships have been suspended and data can no longer be propagated for the devices specified in the E-ConGroup, then I/O is allowed to flow again to the source devices. The log write that was intercepted in this example completes, which allows the dependent data write to be issued to the source site. While these updates are not immediately sent to the remote side, they are propagated through normal SRDF operation once SRDF mirroring is resumed. The time required to perform E-ConGroup processing is a few milliseconds, and may show up as an elongated I/O for in-flight transactions executing during the E-ConGroup trip.
7. Data on remote Symmetrix unit volumes R2(A), R2(C), and R2(E) represent the data on the source Symmetrix units at the instant of the link failure in step 2. From an application and database perspective, the state of the remote data is as if a power failure had happened on the production environment using only the source Symmetrix unit. E-ConGroups maintain data integrity on the remote R2 devices such that databases and applications can be restarted using normal restart processing.

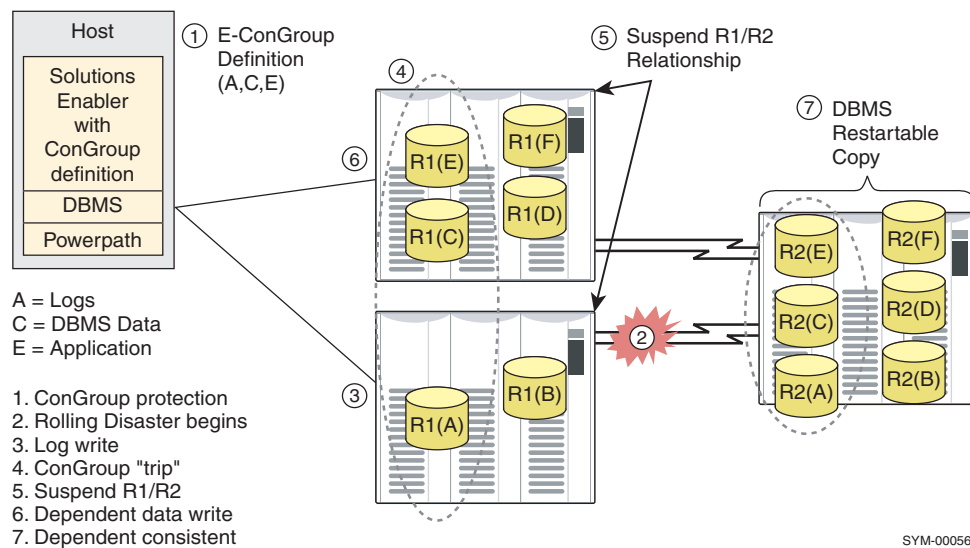


Figure 2-1 ConGroup for UNIX

2.2.3 EMC Change Tracker

Using `symchg` (the Change Tracker command), you can mark areas of Symmetrix disk storage so that logical data objects occupying those areas can be monitored for change.

The command time stamps and marks an area of disk storage occupied by a data object using a DeltaMark (formerly, SDDF) bitmap. This allows a marked area to be examined at a later time to view changes to the stored object.

2.3 PowerPath

EMC PowerPath is host-based software that works with storage systems to intelligently manage I/O paths. PowerPath supports multiple paths to a logical device, the smallest addressable storage unit. Having multiple paths enables PowerPath to:

- ◆ Recover from path failure. PowerPath can automatically detect path failure and redirect I/O to another path.
- ◆ Load balance I/O requests. PowerPath can distribute I/O requests to a logical device across all available paths, thus improving I/O performance and reducing management time and downtime by eliminating the need to configure paths statically across logical devices.

PowerPath operates with several storage systems, on several operating systems, with parallel SCSI or fibre channel data channels.

Although PowerPath provides a great deal of functionality, this document will limit the discussion to PowerPath involvement with EMC Consistency Technology.

This chapter provides information on the following:

◆ 3.1 Overview.....	3-2
◆ 3.2 Understanding the Current Environment.....	3-3
◆ 3.3 Change Tracker Data Collection.....	3-4
◆ 3.4 Planning the Network Configuration.....	3-5
◆ 3.5 Understanding the Characteristics of a Symmetrix Automated Replication Workload	3-7

3.1 Overview

This chapter provides a practical approach to the issue of capacity planning and performance modeling applicable to a Symmetrix Automated Replication solution.

The most frequently asked questions when an installation project begins are:

- ◆ What is the expected cycle time for this asynchronous data replication process?
- ◆ What will happen to the response time of existing applications?
- ◆ Will the storage subsystem support the additional I/O activity created by Symmetrix Automated Replication?

As in any other business, the activity of planning the capacity of a computing system involves obtaining answers to four basic questions:

- ◆ What is the currently installed capacity?
- ◆ What services should be provided in the future?
- ◆ What quality goals are planned for the services (service level agreements)?
- ◆ What is the most cost-effective system configuration to handle current and future services and still meet the planned quality goals?

To help answer the questions above, it is necessary to:

- ◆ Know the current environment
- ◆ Understand the replication workload
- ◆ Collect data for analysis
- ◆ Apply a performance model

3.2 Understanding the Current Environment

Understanding the current environment in this context means understanding the existing or proposed hardware configuration, specifically the Symmetrix storage subsystem and the network configuration.

3.2.1 Existing Hardware Configuration

Every generation of Symmetrix system has a different performance profile. Every different disk device and physical mapping of devices will result in different transfer rates and different cycle times. This information can be collected using the Qualification Survey for Symmetrix Automated Replication Projects. The Qualification Survey is part of the Professional Services that precede this type of implementation. Table 3-1 presents an extraction from this qualification survey.

Table 3-1 Symmetrix Subsystems

Channel Extender (*)	Link Bandwidth Link Type (**)	#of Links	Circuit Distance	FarPoint Buffer	Compression Rate	Disk Emulation (FBA/KD)

(*) Inrange 9800, 9801, CNT USD (model), DWDM (make/model)

(**) T1, ATM OC3, ATM T3, IP, DWDM

Symmetrix Automated Replication creates an additional I/O workload on the storage subsystem. One of the most basic questions to ask when a system will receive additional workload is: Will this system support the increased workload?

To answer this question, we need to know how much of the available system capacity is being used and how much additional workload Symmetrix Automated Replication is going to add to the system. To determine the actual utilization of the storage systems, the following tools are recommended:

- ◆ Host sar-data (statistical analysis reports)
- ◆ EMC Workload Analyzer (EMC ControlCenter™ Workload Analyzer – Product Guide V 4.3 and above)
- ◆ EMC BC Design Tool (EMC Professional Services)

The analysis of the collected data has to determine:

- ◆ Utilization level of logical volumes
- ◆ Utilization level of physical volumes
- ◆ I/Os per second
- ◆ % Writes
- ◆ % Read hits
- ◆ Average response time

It is recommended to work with peak periods when determining the utilization levels.

The next step on this analysis is to determine how much additional workload the replication will add to the existing configuration.

3.3 Change Tracker Data Collection

To help determine how much I/O activity Symmetrix Automated Replication will add to the existing environment, it is necessary to assess incremental changes to the data over discrete periods of time. Change Tracker assists in this task by measuring volume changes, providing information so that informed decisions can be made about the communication line bandwidths and links required for this configuration.

Change Tracker reports help the determination of the locality of reference of the write I/Os and improves the accuracy of the network sizing. When identified that the write I/Os are being driven to the same locality, the requirements for the link bandwidth can be significantly smaller than it would be when calculating link bandwidth requirements based only on write IOs.

ChangeTracker for open systems is available in Solutions Enabler with the DeltaMark license. For more information on the use of ChangeTracker, refer to the Solutions Enabler product guide and the white paper, *Using SYMCLI to Measure Volume Changes with Change Tracker*.

3.4 Planning the Network Configuration

The network configuration is among the most important hardware components to plan and configure. The optimization of the network resources, more specifically link bandwidth, can provide significant cost reductions to the disaster restart solution.

The list in Table 3-2 is also part of the Qualification Survey for Symmetrix Automated Replication Projects. Some information about the network configuration can be obtained during the planning phase. Some information, such as Compression Rate, can be obtained only after the installation of the channel extender and the SRDF connection.

Table 3-2 Planning the Network Configuration

Channel Extender (*)	LinkBandwidth Link Type (**)	#of Links	Circuit Distance	FarPoint Buffer	Compression Rate	DiskEmulation (FBA/CKD)
----------------------	---------------------------------	-----------	------------------	-----------------	------------------	-------------------------

(*) Inrange 9800, 9801, CNT USD (model), DWDM (make/model)

(**) T1, ATM OC3, ATM T3, IP, DWDM

3.4.1 Channel Extenders

There are significant differences among channel extenders. For example, the old Inrange 9800 model does not share the link bandwidth with the Symmetrix RA, while the 9801 does share link bandwidth. This difference can cause a significant change in cycle times. Refer to Table 3-3.

Table 3-3 Comparative Matrix of Channel Extenders

Make / Model	Max # of Ports	Shared or Dedicated Link Bandwidth	Max Compression Rate	Protocol Time	Buffer Size
Inrange 9800	4/2	No sharing. Port less than or equal to 4.5 MB/s.	15:1 (can't utilize > 2:1)	< 0.2 ms per write	1 MB
Inrange 9801	6/4	Shared and striped (writes split on lines).	15:1	< 0.5 – 0.7 ms per write	> 8 MB
CNT USD	8/4	Sharing. No striping.	15:1	2.5 ms per write. (There is a special code that provides < 1.5 ms per write but forces dedicated links.)	> 8 MB

Note: This matrix is just an example. Always consult the provider of the channel extender to obtain current values.

Table 3-4 lists the different link throughputs.

Table 3-4 Link Throughput Summary

Links	T1	E1	T3	E3	ATM-OC3	ESCON	FC
Bandwidth	.15 MB/s	.2 MB/s	4.5 MB/s	3.4 MB/s	15.5 MB/s	17 MB/s	100 MB/s

3.4.2 Circuit Distance

A very common mistake is to assume that the circuit distance and the distance between the sites are the same. The circuit distance can be obtained with the execution of an internal Symmetrix utility (A2,F3). This Symmetrix utility pings the link and provides the echo delay information (i.e., the time to propagate the signal over the link). The time will be presented in milliseconds (ms). Use the minimum ping time as a reference.

To calculate the distance, consider that 1ms represents the round-trip propagation delay for a circuit of 100 km of distance.

3.4.3 FarPoint Buffer

For long distances when using ESCON adapters, it is always recommended to size the FarPoint™ buffers. Only Customer Service Specialists in the CS Performance Group can change FarPoint buffers.

3.4.4 Compression Rate

If the environment already uses a channel extender, it is possible to obtain the actual compression rate. If not, for planning purposes, a conservative approach is recommended. The conservative approach for compression is 2:1.

Note: It is important to investigate if host data compression is used. If it is, we can't expect a significant compression on the channel extender.

3.4.5 Disk Emulation

The disk emulation is either FBA or CKD:

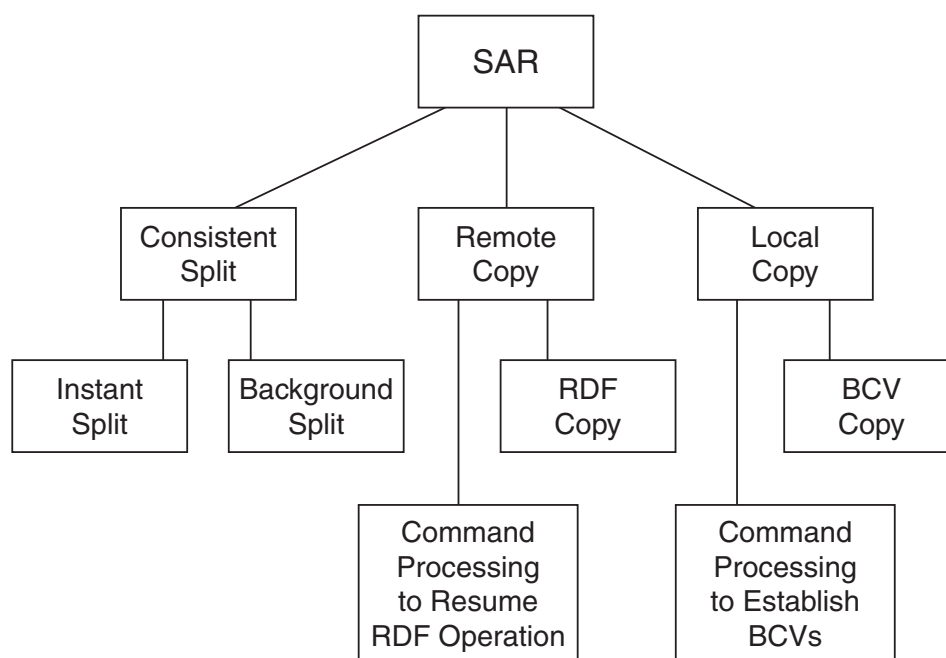
- ◆ FBA — System is configured with 32 KB tracks
- ◆ CKD — System is configured with 56 KB tracks

3.5 Understanding the Characteristics of a Symmetrix Automated Replication Workload

This section provides an overview of the characteristics of a Symmetrix Automated Replication workload. This will aid in the definition of a performance model to calculate the cycles of data replication and determine the additional workload to the logical volumes and physical disks.

3.5.1 Basic Components

A Symmetrix Automated Replication cycle can be partitioned into three basic components: consistent split, remote copy, and local copy. Refer to Figure 3-1.



SYM-000145

Figure 3-1 Symmetrix Automation Replication Workload Components

The consistent split operation is partitioned in instant split and background split.

Instant split is a very fast operation having minimal, if any, affect on the applications. The write I/O is deferred during the instant split operation, which stops the mirroring process between the STD and BCV. If the BCV is to be used locally, the BCV becomes immediately ready for host utilization.

The background split is an asynchronous operation executed in the Symmetrix back end and follows the instant split. The execution time of the background split depends on the number of devices and changed tracks.

The remote copy is partitioned into two components: commands to resume the RDF operation on the BCV/R1–R2 pairs and the copy of the changed tracks from R1 to R2.

The local copy is also partitioned into two components: commands to establish the BCVs to the STDs, and the copy of the changed tracks from STDs to BCVs.

3.5.2 TimeFinder and SRDF Commands

The time to execute the background split, the RDF resume, and the TimeFinder establish commands depends on the number of devices, changed tracks, and number of control units that the SRDF/AR process manages.

EMC professional services have tools to calculate the time to execute the commands, using the data collected in the previous phases of the planning process.

3.5.3 Copy Tasks

Copy tasks are track copies executed by TimeFinder and SRDF. These copy tasks are processed in the Symmetrix back end, i.e., they do not use channel adapters or resources from the hosts.

The time to execute the copy of the track depends on:

- ◆ Number of invalid tracks
- ◆ Distribution of invalid tracks across logical volumes and physical drives

RMC professional services have tools to estimate the throughput rate when executing the copy tasks, using the data collected in the previous phases of the planning process.

For the purpose of creating examples for the next section, *3.5.5 Performance Modeling*, we are going to use 40 MB/s as a conservative reference for internal throughput rate.

Table 3-5 shows the different copy task and related information.

Table 3-5 Copy Task Execution Impacted by Invalid Tracks and Transfer Rate

Copy Task	# of Invalid Tracks	Transfer Rate
BCV Copy	CKD = 56 KB per track FBA = 32 KB per track	Internal transfer rate
RDF Copy	CKD = 56 KB per track FBA = 32 KB per track	External transfer rate Depends on # of RAs / RFs and link bandwidth

3.5.4 Invalid Tracks

After populating remote BCVs for the very first time, all copies executed in a Symmetrix Automated Replication process are *incremental copies* (i.e., just the tracks that were changed during the last cycle of data replication are copied).

The number of invalid tracks and number of write I/Os do not have a 1:1 ratio, so the determination of the number of changed tracks cannot be based only on the identification of the write I/O activity in the hosts. EMC offers a tool called EMC ChangeTracker to improve the accuracy of this analysis.

The internal transfer rate or the transfer rate for a BCV copy is not a fixed value. The transfer rate can vary according to the number of invalid tracks and distribution of invalid tracks across logical volumes and physical disks also as a function of the hardware used and front-end I/O activity. For the purpose of providing a reference for calculations without

knowing the environmental characteristics, we can define two categories of values, listed in Table 3-6.

Table 3-6 BCV Copy Internal Transfer Rates

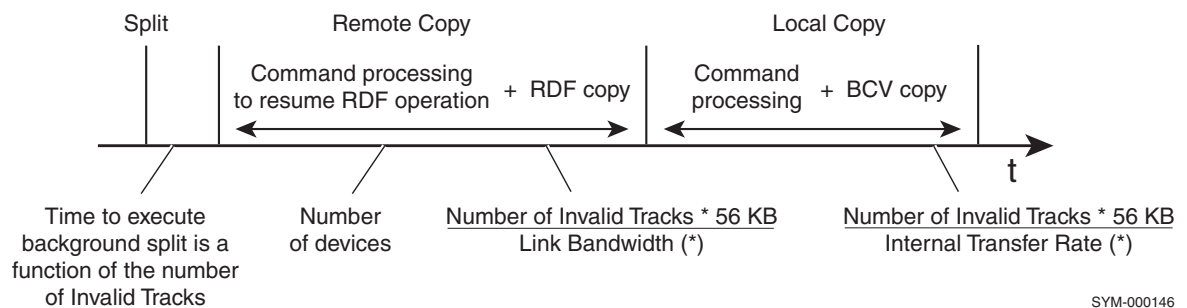
Pessimistic	20 MB/s
Conservative	40 MB/s

To calculate accurate values, it is important to know characteristics of the workload that runs in the storage subsystem.

3.5.5 Performance Modeling

Once the average number of invalid tracks is obtained, the size of the links and the length of the data replication cycle must be determined.

A performance model can be used to analyze different configuration alternatives. This performance model can be created using the components defined in the section 3.5 *Understanding the Characteristics of a Symmetrix Automated Replication Workload* on page 3-7. Figure 3-2 is a sample performance model.



(*) Consult your Systems Engineer to obtain the applicable Internal Transfer Rate for your environment.

Figure 3-2 Low-Level Performance Model

3.5.6 Applying the Model

To apply the model, we can use the following example:

- ◆ 350,000 invalid tracks — 19 GB
- ◆ 120 volumes 3390-3
- ◆ 4 RAs — up to 56 MB/s
- ◆ 1 ATM OC3 — 155 Mb or 15.5 MB/s
- ◆ Channel Extender — Inrange 9801 with a compression rate of 2:1
- ◆ Distance = 0 km

1. Calculate the time to execute the background (BG) split

BG Split Time = # of devices 3390-3 * Average time of a 3390-3 BG split

BG Split Time = 120 * 5 seconds

BG Split Time = 600 seconds or 10 minutes

2. Calculate the time to execute the SRDF Resume command

RDF Resume = # of devices 3390-3 * Average time of a 3390-3 RDF resume/# of concurrent commands defined in SRDF/AR RESUME option

RDF Resume = 120 devices * 5 seconds/4 concurrent commands

RDF Resume = 150 seconds or 2.5 minutes

3. Calculate the time to execute the Remote Copy

RDF Copy = # of invalid tracks * 56 KB/(Link Bandwidth in MB/s * Compression Ratio) / 1024

RDF Copy = 350,000 tracks * 56 KB/(15.5 MB/s * 2) * 1024

RDF Copy = 617.4 seconds or 10.3 minutes

The denominator is limited by the SRDF adapter capacity. For example, if the link bandwidth is larger than the SRDF RA capacity, the RA capacity should be used as the denominator in this function.

4. Calculate the time to execute the TimeFinder (TF) reestablish command

TF Reestablish = #of devices 3390-3 * average time of a 3390-3 TF reestablish/#of concurrent commands defined in SRDF/AR REEST option

TF Reestablish = 120 devices * 5 seconds/4 concurrent commands

TF Reestablish = 150 seconds or 2.5 minutes

5. Calculate the time to execute the Local Copy

TF Copy = # of invalid tracks * 56 KB/Internal Transfer Rate in KB (conservative value)

TF Copy = 350,000 tracks * 56 KB/40,960 KB

TF Copy = 480 seconds or 8 minutes

The 40 MB/s of internal transfer rate for the local copy is a given number considered a conservative value, applicable for this case study. This value should not be used outside the scope of this case study. As said before, the internal transfer rate will be a function of the hardware configuration, volume layouts, I/O activity in the front end of the storage subsystem, etc. Because of this, there is no fixed number to define internal transfer rate for the local copy.

EMC recommends discussing this transfer rate with the EMC performance specialist that supports the account or work with EMC professional services.

6. Calculate the total cycle time

Summing up all the components of the cycle time, we get to the result of 33 minutes to execute the data replication cycle. Refer to Figure 3-3.

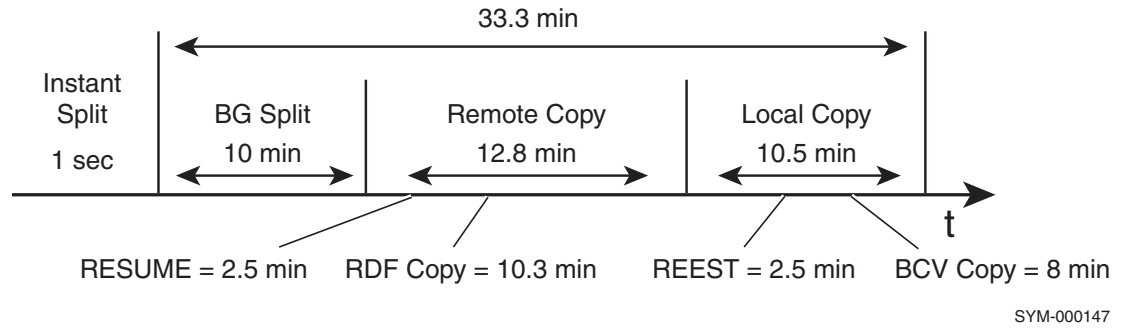


Figure 3-3 Calculate Total Cycle Time

At this moment, as part of the performance modeling, different hardware configurations should be modeled with the purpose of providing different price/performance configurations.

3.5.7 Considerations for Performance Modeling

The following are considerations for performance modeling.

3.5.7.1 TimeFinder Establish Times

Discuss the recommended internal transfer rate for TimeFinder establish operations with a performance specialist. Different hardware configurations like Symmetrix family and disk types will present different values. Front-end I/O activity can also affect the internal transfer rates.

3.5.7.2 SRDF Scalability

When sizing a data replication solution, it is important to consider the effective capacity of SRDF. Remember that different Symmetrix families and different versions of Symmetrix Engenuity will affect SRDF throughput rate. As a reference, in a Symmetrix 8000 series, the maximum throughput rate of an RA (ESCON) group is 72 MB/s scaling up to six ESCON adapters, and the maximum throughput rate of an RF group (fibre) is 120 MB/s scaling up to four remote fibre adapters.

Note that even though SRDF can scale to a very large throughput rate, achieving the maximum throughput rate depends on several factors such as channel extender, link bandwidth, distance, emulation type (FBA/CKD), and distribution of I/Os on logical devices.

Users can obtain detailed information about SRDF scalability from EMC personnel with access to the internal EMC SPEED website.

This chapter presents information on the following topics:

◆ 4.1 Overview.....	4-2
◆ 4.2 Practical Uses.....	4-3
◆ 4.3 Consistent Split	4-4
◆ 4.4 Creating a Local DBMS-Restartable BCV Copy	4-7
◆ 4.5 Creating a DBMS-Restartable BCV Copy on a Remote Symmetrix	4-8
◆ 4.6 Creating DBMS-Restartable Copies on Both Sides	4-9
◆ 4.7 Using Consistent Split When Performing SRDF/AR.....	4-10
◆ 4.8 Performing a Split with Database Assist.....	4-11

4.1 Overview

This chapter describes TimeFinder functionality included in versions of EMC Solutions Enabler up to version 5.3 running on Symmetrix units using Enginuity Version 5x66, 5x67, 5568, and 5669. This chapter describes the TimeFinder consistent split operation for obtaining a local or remote DBMS-restartable BCV copy of the database. This chapter also describes how you can use consistent split with Symmetrix Automated Replication (SRDF/AR).

4.2 Practical Uses

When databases become very large (larger than one terabyte), the time and resources required for host-based software to protect or run decision-support queries on these databases becomes a critical factor. The time required to quiesce or shut down the database for offline operations is no longer acceptable for most database applications. When you need to perform such database operations, the use of TimeFinder consistent split allows you to split off a consistent, DBMS-restartable copy of the database in seconds with no interruption to online service.

TimeFinder software with PowerPath support provides a consistent-split implementation of instant split that allows you to split off a consistent, DBMS-restartable BCV copy of your database without having to quiesce or shut down the database first. The impact on your production environment is negligible. Also, beginning with EMC Solutions Enabler Version 5.1, TimeFinder can use Enginuity Consistency Assist (ECA) to perform consistent split across multiple, heterogeneous hosts *without* PowerPath support.

Consistent split can also play a role in Symmetrix Automated Replication, a software feature that allows you to set up remote mirroring to be automatic according to your own predefined copy cycle (every hour, for example). You can use a single-hop disaster restart configuration that permits controlled data loss (at most, one copy cycle). If greater protection is required, a multihop restart configuration can provide synchronous disaster restart with zero data loss (that is, all completed transactions on the source are completed on the target). Using TimeFinder consistent split within a `symreplicate` session to copy data to remote sites provides consistency and automation benefits for successful restart solutions.

Compared to traditional disaster *recovery* solutions with their long restart time and high data loss, disaster *restart* solutions using these Symmetrix features provide remote restart with a short restart time and low data loss.

For more information about SRDF/AR, refer to the white paper *Using SYMCLI to Perform Symmetrix Automated Replication* (P/N 300-000-078).

4.3 Consistent Split

Instant split reduces the wait period during a split to a short time by decomposing the splitting process into two phases: a *foreground split* and a *background split*. During an instant split, the system executes the foreground split almost instantaneously and returns a successful status to the host. This action allows I/O operations to the standard and BCV to continue immediately. At the same time, the background split continues to split the BCV pair until the split is complete.

Note: Beginning with the TimeFinder component of EMC Solutions Enabler Version 5.0 running on Symmetrix units using Enginuity Version 5568, all splits are instant splits. If you include the `-instant` option with the command `symmir split`, SYMCLI returns immediately after the foreground split. Omitting the `-instant` option causes SYMCLI to wait until the background split completes before returning.

Consistent split is a special implementation of instant split that allows you to avoid inconsistencies and restart problems that can occur if you split a database-related BCV without first quiescing the database. The difference between a normal instant split and a consistent instant split is the addition of PowerPath or ECA to suspend the database device writes at the host level for a very short time while the foreground split occurs.

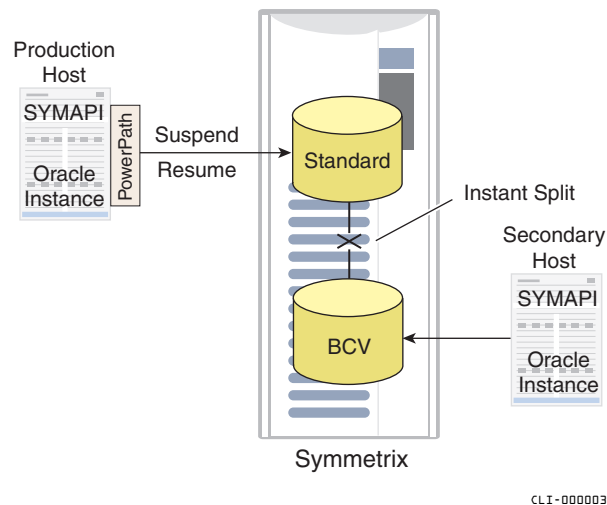


Figure 4-1 Consistent Split on a Local Symmetrix

Figure 4-1 illustrates the use of local consistent split with a database management system (DBMS). When you issue a `symmir split` command with an ECA- or PowerPath-related option from the production host, I/O is temporarily suspended during the following sequence:

1. The Symmetrix API (SYM API) identifies the standard devices that hold the database.
2. SYM API validates that all identified BCV pairs can be split.
3. SYM API sends a "suspend I/O" log message to PowerPath or ECA.
4. PowerPath or ECA suspends I/O to the standard devices that hold the database.
5. The DBMS cannot write to the devices and subsequently waits for these devices to become available before resuming any further I/O. The device I/O is usually suspended for less than one second.

6. SYMAPI sends an instant split request to all BCV pairs in the specified device group and waits for the Symmetrix to acknowledge that the foreground split has occurred.
7. SYMAPI sends a “resume I/O” message to PowerPath or ECA.
8. The DBMS resumes writing to the standard devices that hold the database.

The BCV devices now contain a DBMS-restartable copy of the database that is consistent with the production database up until the time of the instant split. The production DBMS instance (on the production host in Figure 1) has no knowledge that the split or suspend/resume operation occurred. When a DBMS instance on the secondary host is started against the split BCVs, there is no record of a successful shutdown or backup. Therefore, the secondary DBMS instance views the BCV database copy as if it had crashed and proceeds to perform the normal crash recovery sequence in starting the database copy.

Before performing a consistent split, it is a good practice to issue a checkpoint from the production database. A database checkpoint flushes dirty pages out to disk and minimizes recovery on the target database.

As mentioned previously, beginning with EMC Solutions Enabler Version 5.1, you can use ECA to perform consistent splits on BCV pairs across multiple, heterogeneous hosts *without* PowerPath support. To do so, however, requires that you have either a control host (with no database) or a database host with a dedicated channel from which to perform the control. Figure 4-2 on page 4-6 shows a control host that can perform ECA consistent splits for three database hosts that access devices on the Symmetrix unit.

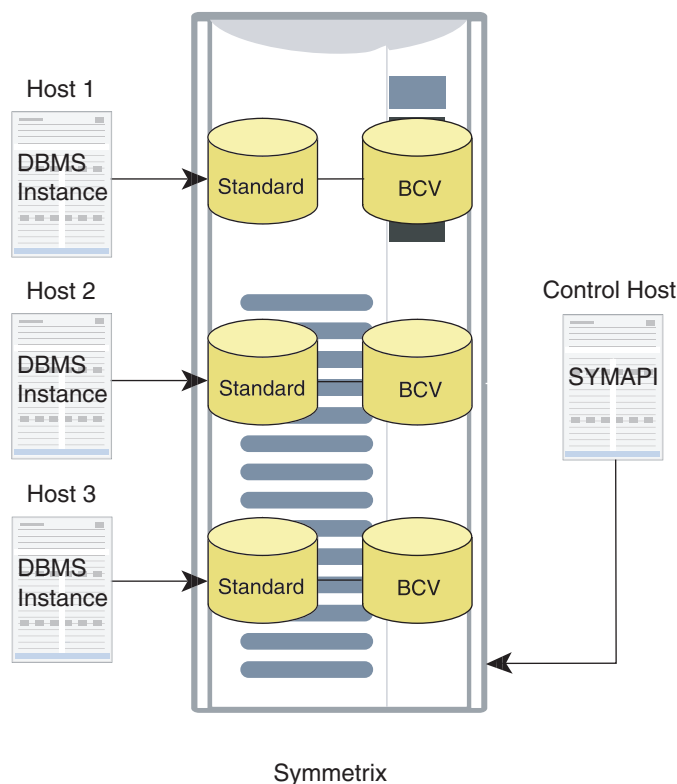


Figure 4-2 Performing an ECA Consistent Split across Multiple Heterogeneous Hosts

The nature of the ECA consistent split depends on the device groups that you define on the control host. For example, if you define a device group that includes all the devices being accessed by the three database hosts, then you can split all BCV pairs related to those hosts at the same time. If you define a device group that includes just the devices being accessed by Host 1, then you can split those BCV pairs related to Host 1 without affecting the other hosts. Similarly, you can define a device group on the control host for the Host 2 devices and another device group for the Host 3 devices.

Table 4-1 lists differences between the ECA `-consistent` option and the PowerPath `-ppath` option.

Table 4-1 Performing Consistent Splits Using ECA `-consistent` or PowerPath `-ppath`

	<code>-consistent</code>	<code>-ppath</code>
Minimum Enginuity support version	5x67	5x66
Minimum Solutions Enabler support version	5.1	4.3.1 ^a
Multihost support	Yes	No
Available on operating systems where PowerPath does not run ^b	Yes	No
Can be run from a host not running the DBMS	Yes	No
Requires a control host (with no database) or a host with a dedicated channel	Yes	No

a. With PowerPath Version 2.0 software (or higher).

b. For currently supported operating systems, refer to the *Solutions Enabler Release Notes*.

4.4 Creating a Local DBMS-Restartable BCV Copy

The following general setup¹ for creating consistent BCV copies of a database locally is similar to setting up for other TimeFinder operations:

1. Create a Regular type device group.

```
symdmg create ProdBgrp -type regular
```

2. Add to the device group those standard devices that hold the database (PowerPath-connected devices if you are using PowerPath support).

```
symld -g ProdBgrp addall dev -range 286:28B
```

3. Associate with the device group those BCV devices that will hold the restartable copy of the database.

```
symbcv -g ProdBgrp associateall dev -range 39A:39F
```

4. Fully establish all BCV pairs in the device group.

```
symmir -g ProdBgrp establish -full -noprompt
```

When the BCV pairs of a device group (ProdBGrp, for example) are synchronized, perform the consistent split using one of several options to the `symmir split` command:

By providing a database type (`oracle`, `sqlserver`, `ibmudb`, or `informix`). This operation requires setting the appropriate environment variables first to allow SYMCLI to log in to the database:

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

SYMAPI queries the Oracle database manager for the location of those devices that hold the database. PowerPath or ECA suspends I/O to the devices identified by the Oracle database manager. This option may be expensive in terms of the time required to discover the devices.

By providing the physical device names of PowerPath-connected standard devices. In this case, you are responsible for specifying the correct list of standard devices that hold the database. For example, to specify PowerPath devices whose physical device names are `c1t0d0`, `c1t0d1`, and `c1t0d2`:

```
symmir -g ProdBgrp split -instant -ppath /dev/rdisk/c1t0d0 \
                                         /dev/rdisk/c1t0d1 /dev/rdisk/c1t0d2
```

When the devices that hold the database are the only standard devices in the device group, it is simpler to substitute the `stddevs` keyword for the list of device names. For example:

```
symmir -g ProdBgrp split -instant -ppath stddevs
```

Beginning with SYMCLI version 5.1, by providing the `-consistent` option to perform a consistent split on all BCV pairs in the device group. For example:

```
symmir -g ProdBgrp split -consistent
```

By providing a list of VxFS file system names (such as `ora1`, `ora2`, `ora3`). This operation suspends I/O to these three file systems during the instant split operation but does not involve PowerPath:

```
symmir -g ProdBgrp split -instant -vxfs /ora1 /ora2 /ora3
```

1. An alternative method of creating this device group would be to use `symrdb db2dmg` to automatically discover the standard devices that hold the database. However, you would still need to add the BCVs to the device group as described in step 3.

4.5 Creating a DBMS-Restartable BCV Copy on a Remote Symmetrix

Using SRDF technology and the TimeFinder RDF flag (the `-rdf` option), you can establish a BCV pair on the remote Symmetrix unit (Site B in Figure 4-3). By doing this, the remotely associated BCV is synchronized with the target (R2) device until such time that you may decide to perform a consistent split to create a DBMS-restartable database copy on the BCVs of a remote Symmetrix unit.

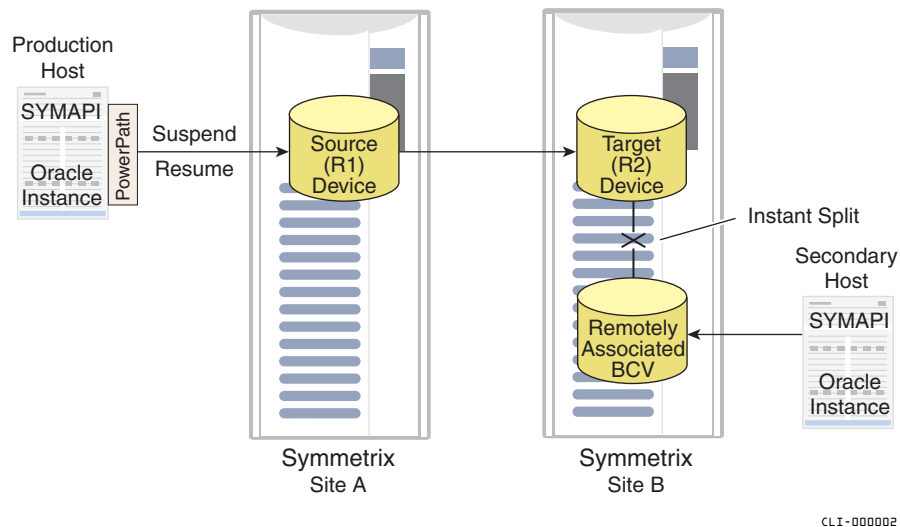


Figure 4-3 Consistent Split on a Remote Symmetrix

To perform a consistent split successfully in an SRDF environment, you must run SRDF in synchronous mode. The following steps outline the procedure for performing a consistent split on a remote BCV pair:

1. Create an RDF1 type device group. For example:

```
symdbg create Rdf1Grp -type rdf1
```
2. Add to the device group those R1 standard devices that hold the database (PowerPath-connected devices if using PowerPath support).

```
symld -g Rdf1Grp add dev 27C
```
3. Determine which BCV devices on the remote Symmetrix unit (sid 505, for example) are not already part of a device group and are free to add to your device group.

```
symdev list -sid 505 -bcv
```
4. Remotely associate with the device group BCV devices that will hold the restartable database copy.

```
symbcv -g Rdf1Grp associate dev 090 -rdf
```
5. Fully establish all BCV pairs in the device group.

```
symmir -g Rdf1Grp establish -full -rdf
```
6. Perform a consistent split of all PowerPath-connected standard devices with their remotely associated BCVs.

```
symmir -g Rdf1Grp split -rdf -instant -ppath stddevs
```

4.6 Creating DBMS-Restartable Copies on Both Sides

Figure 4-4 illustrates how you can copy the same data to BCV devices on two Symmetrix units. By performing a consistent split operation from the production host using the `-both_sides` option, you can split the BCV pairs on both sides at the same time. If the R1 source and R2 target devices are paired with concurrent BCVs, the consistent split is applied to four BCVs. The BCVs on both sides then contain data that is consistent with the R1 source data up to the time of the split and are accessible to their hosts.

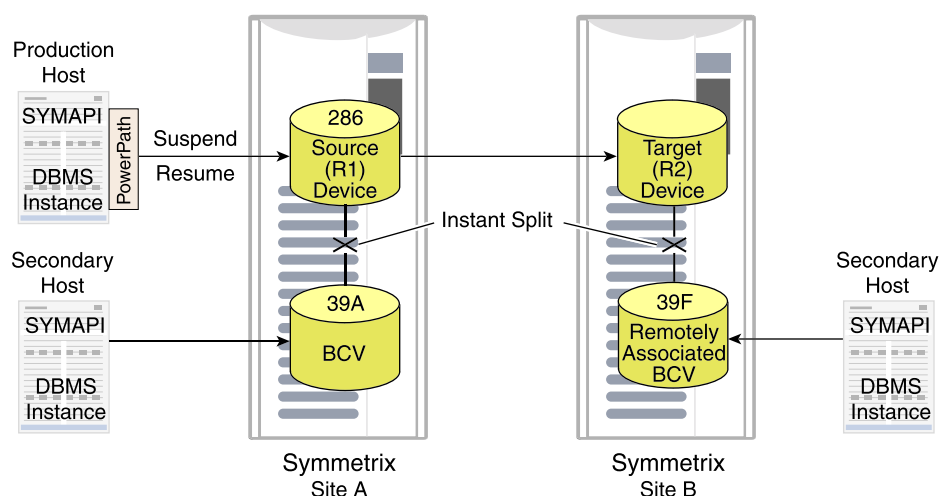


Figure 4-4 Copying the Same Data on Local and Remote Symmetrix Units

The following steps outline an example of copying the same data to BCVs on two Symmetrix units:

1. Create an RDF1 type device group.

```
symdmg create Rdf1Grp -type rdf1
```
2. Add to the device group the R1 standard device (286) on the local Symmetrix (sid 3264) that will hold the production data (a PowerPath-connected device if using PowerPath support).

```
symld -g Rdf1Grp -sid 3264 add dev 286
```
3. Associate with the device group the BCVs on both Symmetrix units.

```
symbcv -g Rdf1Grp -sid 3264 associate dev 39A
symbcv -g Rdf1Grp -sid 3265 associate dev 39F -rdf
```
4. Establish the BCV pairs on both Symmetrix units. DEV001 is the logical name of the PowerPath device (286). The first command establishes the local BCV pair at Site A. The second command using the `-rdf` option establishes the remotely associated BCV at Site B with the R2 standard device there.

```
symmir -g Rdf1Grp establish -full DEV001 -noprompt
symmir -g Rdf1Grp establish -full DEV001 -rdf -noprompt
```
5. When the BCV pairs are fully synchronized, perform a consistent split on both BCV pairs. Specify the `-ppath` option with either the `stddevs` keyword (all PowerPath devices in the device group) or a PowerPath device name (for example, `/dev/rdisk/emcpower1c`).

```
symmir -g Rdf1Grp split -instant -both_sides -ppath stddevs
```

4.7 Using Consistent Split When Performing SRDF/AR

You can use the SRDF and TimeFinder components of SYMCLI to automate data copies across RDF links with Symmetrix Automated Replication (SRDF/AR). When used with the TimeFinder consistent split option, SRDF/AR ensures that a remotely associated BCV always contains a DBMS-restartable copy of the data. In the SRDF/AR single-hop configuration (Figure 4-5), PowerPath suspends I/O from the production host when a consistent split is invoked for the local BCV pair (step 1), and resumes I/O while the split-off data is copied to the remote site (steps 2 and 3).

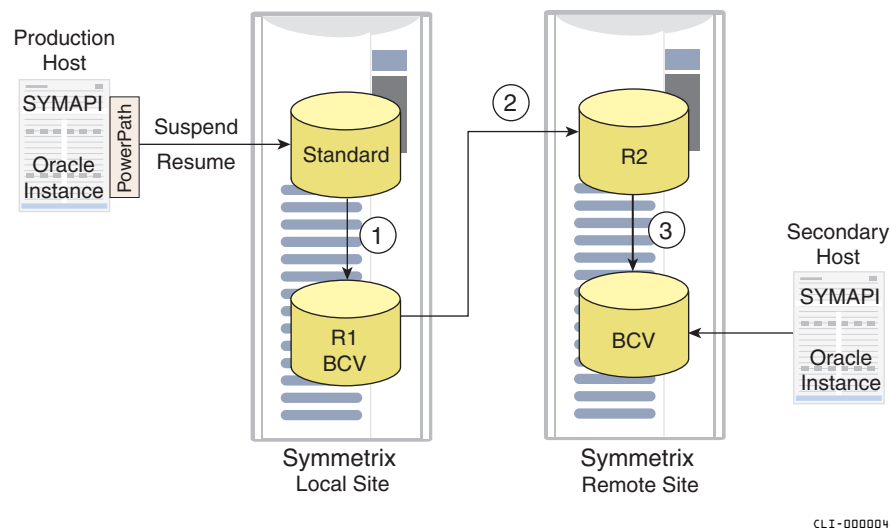


Figure 4-5 Automated Data Copying in the SRDF/AR Single-Hop Configuration

In the multihop SRDF/AR configuration (Figure 4-6), PowerPath suspends I/O from the production host when a consistent split is invoked for the remote BCV pair (step 2), and resumes I/O while the split-off database copy is propagated to a BCV at a third site (steps 3 and 4).

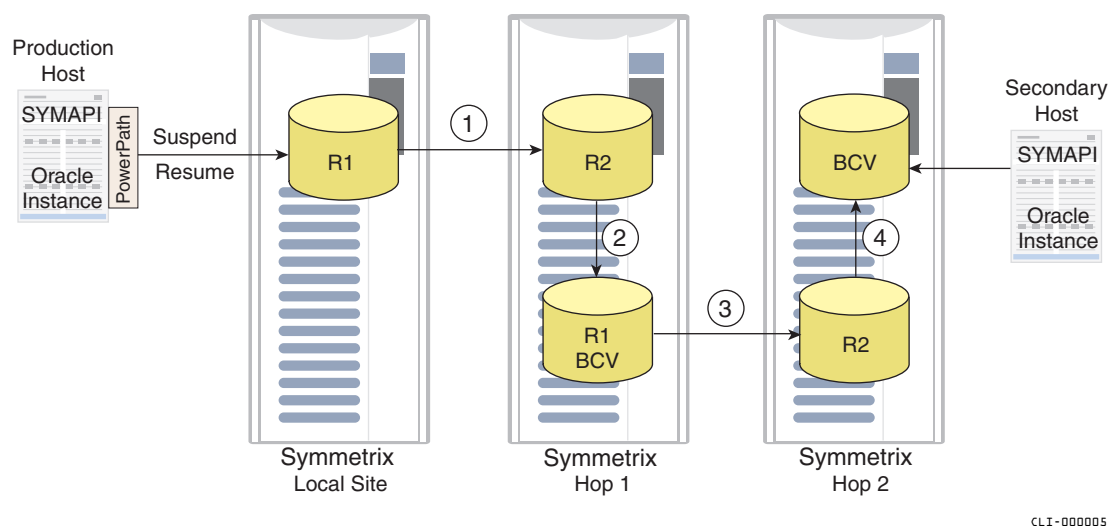


Figure 4-6 Automated Data Copying in the SRDF/AR Multihop Configuration

For more information about performing SRDF/AR operations, including examples, refer to the white paper *Using SYMCLI to Perform Symmetrix Automated Replication* (P/N 300-000-078).

4.8 Performing a Split with Database Assist

In database environments where more than one BCV pair is being split at the same moment, typical TimeFinder instant split operations are performed in an unpredictable order because of the unpredictable nature of I/O load on each device. If you split a database without quiescing it first, it is possible to end up with BCV devices holding user data that is more recent than BCV devices holding log data.

Performing splits without PowerPath or ECA is similar for most supported databases. If you do not have PowerPath or EMC Solutions Enabler Version 5.1 (or higher) installed, you can use the `symioctl` command to place your database in backup mode (for Oracle) or momentarily freeze the database (such as Informix, SQL Server, etc.), suspending any database updates being written to the disk for the time it takes for the foreground split to occur. When the foreground split completes, use the `symioctl` command again to resume access to the database. Keep in mind, however, that the consistent split described in previous sections circumvents this operation and is a cleaner and safer split on the BCV database environment.

For SYMCLI to access a database, set the `SYMCLI_RDB_CONNECT` environment variable and those database application variables that are relevant for your database system. The following example sets the `SYMCLI_RDB_CONNECT` 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
```

Before beginning any SYMCLI database command session, set the appropriate database application variables on the host where the database instance (or server) is running (refer to Table 4-2). For example, setting `ORACLE_HOME` specifies the location of the Oracle binaries. Setting `ORACLE_SID` specifies the database instance name:

```
export ORACLE_HOME=/disks/symapidvt/oraclehome/ap1179
export ORACLE_SID=ap1179
```

Table 4-2 Defining Environment Variables for Database Applications

For Database	Set Variables
Oracle	ORACLE_HOME and ORACLE_SID.
Informix Dynamic Server	INFORMIXDIR, ONCONFIG, and INFORMIXSERVER.
Sybase	SYBASE and DSQUERY . If the OCS - 12_0 directory is not located in the \$SYBASE directory, set LD_LIBRARY_PATH, LIBPATH, or SHLIB_PATH, depending on whether you are running SunOS, AIX, or HP-UX.
IBM DB2/UDB	DB2INSTANCE. Set LD_LIBRARY_PATH, LIBPATH, or SHLIB_PATH, depending on whether you are running SunOS, AIX, or HP-UX.
SQL Server; Exchange	None required.

SQL Server and Exchange do not have variables that specify a server instance. Beginning with EMC Solutions Enabler Version 5.1, SYMCLI supports two or more named SQL Server instances running on the same Windows 2000 box. To connect to an SQL server instance, add a service name (an ODBC system DSN) to the `RDB_CONNECT` variable:

```
set SYMCLI_RDB_CONNECT=username/password@service
```

To create this service name (ODBC system DSN), click **Control Panel, Administrative Tools, Data Sources (ODBC)**, and the **System DSN** tab. Create an ODBC system DSN. When prompted to choose a data source from a list of sources, choose “SQL Server.” When prompted for the named instance of **SQL Server**, enter the instance name.

4.8.1 Freezing an Oracle Database

The following commands place the Oracle database in hot backup mode, freeze the database, perform an instant split on all BCV pairs in a device group named ProdBgrp, thaw the database, and end backup mode.

```
symioctl begin backup -type oracle
symioctl freeze -type oracle
symmir -g ProdBgrp split -instant -noprompt
symioctl thaw -type oracle
symioctl end backup -type oracle
```

This operation on an Oracle database does not directly affect metadata I/O. The operation does not affect any Oracle files that are not related to this database instance or any non-Oracle files in the same file system. Consequently, freeze-and-thaw alone does not guarantee Oracle database integrity as consistent split does. However, this operation does enhance the performance of splits if you are unable to use Enterprise TimeFinder Consistent Split.

4.8.2 Freezing an Informix Database

The following commands freeze an Informix database (-type is informix), perform an instant split on all BCV pairs in a device group named ProdBgrp, and thaw the database.

```
symioctl freeze -type informix
symmir -g ProdBgrp split -instant -noprompt
symioctl thaw -type informix
```

Once the foreground split completes (usually a fraction of a second), the thaw action resumes full I/O access to the database.

4.8.3 Freezing an SQL 2000 Database

Snapshot commands for SQL Server 2000 and higher support BACKUP and RESTORE database WITH SNAPSHOT operations using the Virtual Device Interface (VDI). The symioctl begin snapshot command sends a “BACKUP DATABASE TO VIRTUAL_DEVICE WITH SNAPSHOT” SQL command to SQL Server, beginning the snapshot backup and suspending writes for the specified database.

The symioctl begin snapshot command suspends writes to a database named mydb and creates a savefile named mydb.sav. Then an instant split is performed on BCV pairs in the device group.

```
symioctl begin snapshot -type sqlserver mydb savefile c:\temp\mydb.sav
symmir -g ProdBgrp split -instant -noprompt
symioctl end snapshot -type sqlserver mydb
```

The symioctl end snapshot command causes the “BACKUP DATABASE” SQL command for the mydb database to complete. Writes to the database can resume, and the snapshot metadata is saved to the savefile. The metadata from the savefile is used by SQL Server to logically restore the database if needed (by using a subsequent symioctl restore snapshot command). For example:

```
symioctl restore snapshot -type sqlserver mydb savefile c:\temp\mydb.sav
```

4.8.4 Freezing an IBM UDB Database

The following commands freeze an IBM UDB database (`-type is ibmudb`), perform an instant split on all BCV pairs in a device group named `ProdBgrp`, and thaw the database.

```
symioctl suspend -type ibmudb
symmir -g ProdBgrp split -instant -noprompt
symioctl resume -type ibmudb
```

4.8.5 Freezing a Sybase Database

The following sequence freezes a Sybase database (`-type is sybase`), performs an instant split on all BCV pairs in a device group named `ProdBgrp`, and thaws the database.

```
symioctl suspend -type sybase
symmir -g ProdBgrp split -instant -noprompt
symioctl resume -type sybase
```

This chapter provides information when using SYMCLI:

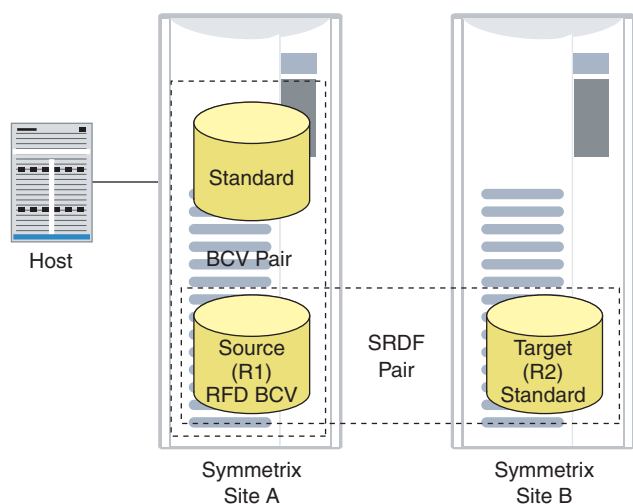
- ◆ 5.1 Overview5-2
- ◆ 5.2 Establishing a BCV Pair with an RDF BCV5-3
- ◆ 5.3 Copying Data to a Remotely-Associated BCV5-4
- ◆ 5.4 Copying Data in a Complex Multihop Environment5-6
- ◆ 5.5 Using Symmetrix Automated Replication5-10
- ◆ 5.6 Accessing a Concurrent BCV while Running SRDF/AR5-19

5.1 Overview

This chapter describes SRDF and TimeFinder functionality included in versions of EMC Solutions Enabler up through version 5.3 running on Symmetrix units using Enginuity versions 5x66, 5x67, 5568, and 5669. While this paper discusses the implementation of SRDF/AR on open systems platforms, similar functionality exists on the mainframe platform. For details of the implementation of SRDF/AR on the mainframe platform, refer to *EMC TimeFinder OS/390 and z/OS Product Set Product Guide*.

5.2 Establishing a BCV Pair with an RDF BCV

With both SRDF and TimeFinder installed, EMC can configure an SRDF device either as an RDF standard device or an RDF BCV device. An *RDF BCV* device is a BCV that is configured in a one-to-one relationship with a remote target (R2) device that mirrors the BCV, forming an SRDF pair as shown in the illustration. You can also establish an RDF BCV device as part of a BCV pair (shown in Figure 5-1 on the Site A Symmetrix system) but, in doing so, the RDF BCV is suspended from copying data to its target (R2) device.



CLI-000089

Figure 5-1 Distinguishing between an SRDF Pair and an RDF BCV Pair

To copy data from the standard device to the RDF BCV device at Site A, you can build a Regular device group (RegGrp, for example) on the local host that includes these two devices. For example:

```
symdg create RegGrp -type regular
symld -g RegGrp add dev 020
symbcv -g RegGrp associate dev 090
```

Use the `symmir establish` command to initiate their synchronization:

```
symmir -g RegGrp establish -full
```

When established as a BCV pair, the RDF BCV is suspended from copying data to its remote (R2) target. To resume mirroring between them, you can split the BCV pair using the `-remote` option, which begins copying data automatically to the R2 side, or you can perform a normal split and re-establish the RDF BCV with its remote (R2) target device as shown in the following example:

```
symmir -g RegGrp split
symrdf -g RegGrp establish -bcv
```

The `symrdf establish` command with the `-bcv` option resumes the RDF links and initiates the propagation of data from the source RDF BCV device to its remote (R2) target device.

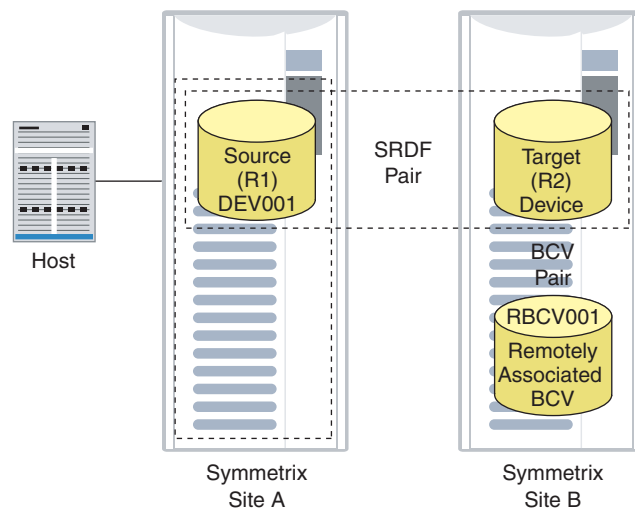
If later you want to reestablish the BCV pair incrementally, perform a `symmir establish`. If, instead, you want to resynchronize the BCV pair but restore the standard device with data from the BCV, use a restore operation. In either case, the RDF BCV is again suspended from copying data to its remote (R2) target.

```
symmir -g RegGrp restore
```

5.3 Copying Data to a Remotely-Associated BCV

When EMC installs an SRDF configuration, the installers usually establish the SRDF pairs at that time. Under normal circumstances, an SRDF pair is usually synchronized.

A BCV pair, on the other hand, is a relationship that you create using the `symmir` command, matching a device configured as a BCV with a specific standard device. Using SRDF technology and the TimeFinder RDF flag (`-rdf` option), you can establish a BCV pair on the remote Symmetrix system (Site B in Figure 5-2). By doing this, the remotely-associated BCV is synchronized with the SRDF remote (R2) mirror until such time that you may decide to split the BCV pair.



CLI-000090

Figure 5-2 Copying Data across RDF Links to a Remotely-Associated BCV

To copy data from the source (R1) standard device to the remotely-associated BCV device, you can build an RDF1 device group (Rdf1Grp, for example) on the local host that includes these two devices (for example, standard device 27C from Site A and BCV device 2E9 from the Site B). The `-rdf` option is required when targeting the remotely-associated BCV for inclusion in the device group (if the source device is an R1 BCV instead of a standard device (as in Figure 4), include the `-bcv` option also with the `symbcv associate` command):

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp add dev 27C
symbcv -g Rdf1Grp associate dev 2E9 -rdf
```

When establishing the remote BCV pair, assume that the source (R1) standard device and its remote (R2) target device are already synchronized, as is usually the case. A `symmir establish -rdf` command from the local host establishes the remote BCV pair, copying data from the standard's remote R2 mirror to the remotely-associated BCV:

```
symmir -g Rdf1Grp establish -full -rdf
```

Once the establish operation is complete, any new data written to the source (R1) standard device is copied to its remote (R2) mirror and subsequently to the remotely-associated BCV.

The `symmir split -rdf` command splits the remote BCV pair, making the remotely-associated BCV available to its host for business continuance tasks or a restore operation, if needed:

```
symmir -g Rdf1Grp split -rdf
```

If you want to reestablish the remote BCV pair later, perform an incremental establish by issuing the `symmir establish -rdf` command without the `-full` option. If, instead, you need to restore data from the remotely-associated BCV back to the local R1 standard device, the process involves two steps:

1. Copy the data from the remotely-associated BCV to its paired R2 standard device. For example:

```
symmir -g Rdf1Grp restore -rdf
```

2. Then copy that data from the R2 standard device back to the R1 standard device. For example:

```
symrdf -g Rdf1Grp restore
```

Once the data transfer initiates successfully in step 1, you can begin the data copy to the R1 standard in parallel, even though the BCV copy to its R2 standard is still in progress. Once the *restore* operation is complete, any new data written to the R1 standard is copied to the remotely-associated BCV device while this remote BCV pair exists. *EMC recommends this two-step process instead of the one-step process that uses `symmir restore -remote`. The two-step process is safer and less error prone.*

5.4 Copying Data in a Complex Multihop Environment

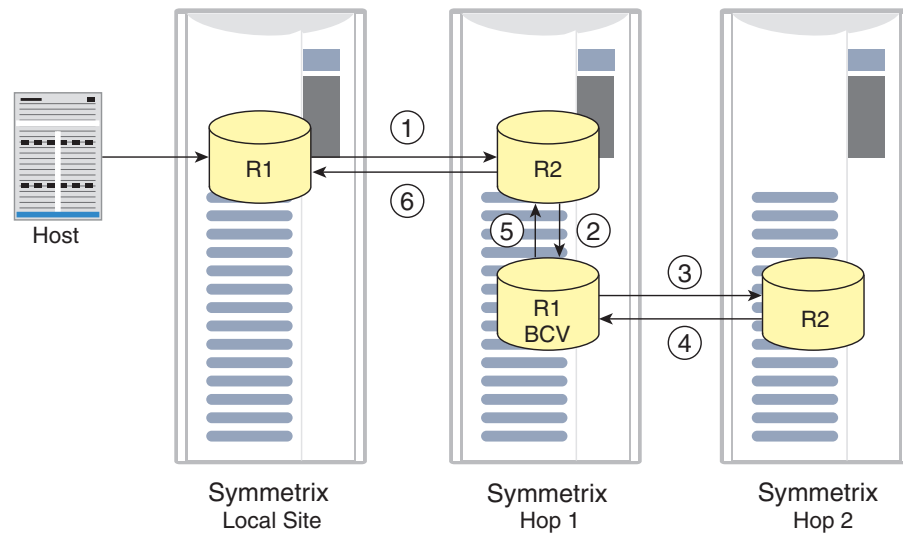
TimeFinder and SRDF components of SYMCLI allow you to manage complex remote configurations from your local host. One of the basic remote configurations is to have a remote site (Hop 1) serve as a remote mirror to the standard devices on a local Symmetrix unit. Multihop configurations include having a third site (Hop 2) serve as a remote mirror to the remotely-associated BCV devices at Hop 1.

Targeting commands to various multihop devices and links in an SRDF multihop environment requires an understanding of several `symrdf` and `symmir` commands and their options. The target of an SRDF or TimeFinder operation is determined by the option or options that you choose with these commands.

Keep in mind that `symrdf` commands perform control operations on SRDF pairs. TimeFinder `symmir` commands perform control operations on BCV pairs, regardless of whether the BCV pair is local or remote. Also, you create the device group on the host attached to the local site Symmetrix.

5.4.1 Targeting Remote Devices when the Data Source Is the Local Standard RDF Device

The following commands issued from the local host show the use of various options to copy data across RDF links to targets associated with the local standard device. Steps 1, 2, and 3 involve copying data from the local Symmetrix to the Hop 2 Symmetrix. Steps 4, 5, and 6 involve restoring data from the Hop 2 Symmetrix to the local Symmetrix. An RDF1 device group named `Rdf1Grp` has been created and the appropriate devices added to it (see the section 5.3 *Copying Data to a Remotely-Associated BCV* on page 5-4).



CLI-000091

Figure 5-3 Copying Data to Remote Sites from a Local Standard Device

1. When EMC installs an SRDF configuration, the installers usually establish SRDF pairs at that time. Assume here that the local R1 standard device is synchronized with its remote R2 mirror. To prepare for a restore operation later, the `symrdf split` command splits the local SRDF pair.

```
symrdf -g Rdf1Grp split
```

2. The `symmir establish` command copies data from the R2 mirror to its paired BCV. Using the `-rdf` option in the command targets the remotely-associated BCV. The `symmir verify -i` command checks every 30 seconds until the BCV pair is synchronized. The `symmir split` command splits the Hop 1 BCV pair.

```
symmir -g Rdf1Grp establish -rdf
symmir -g Rdf1Grp verify -rdf -i 30
symmir -g Rdf1Grp split -rdf
```

3. While part of a BCV pair, the Hop 1 BCV is temporarily suspended from copying data to its remote R2 target. To resume mirroring between the two, you can split the Hop 1 BCV pair (as was done in step 2) and re-establish the Hop 1 BCV now with its remote R2 mirror. The `-rbcv` option targets the Hop 2 device. The `symrdf verify -i` command checks every 30 seconds until the remote SRDF pair is synchronized. The `symrdf split` command splits that SRDF pair. The copy on Hop 2 can now be used for business continuance tasks or a restore operation, if needed.

```
symrdf -g Rdf1Grp establish -rbcv
symrdf -g Rdf1Grp verify -rbcv -i 30
symrdf -g Rdf1Grp split -rbcv
```

4. The `symrdf restore` command restores the Hop 1 BCV from its Hop 2 remote mirror. The next `symrdf split` command splits that SRDF pair, protecting the integrity of the Hop 2 copy.

```
symrdf -g Rdf1Grp restore -rbcv
symrdf -g Rdf1Grp verify -rbcv -i 30
symrdf -g Rdf1Grp split -rbcv
```

5. The `symmir restore` command restores the Hop 1 R2 mirror from the Hop 1 BCV.

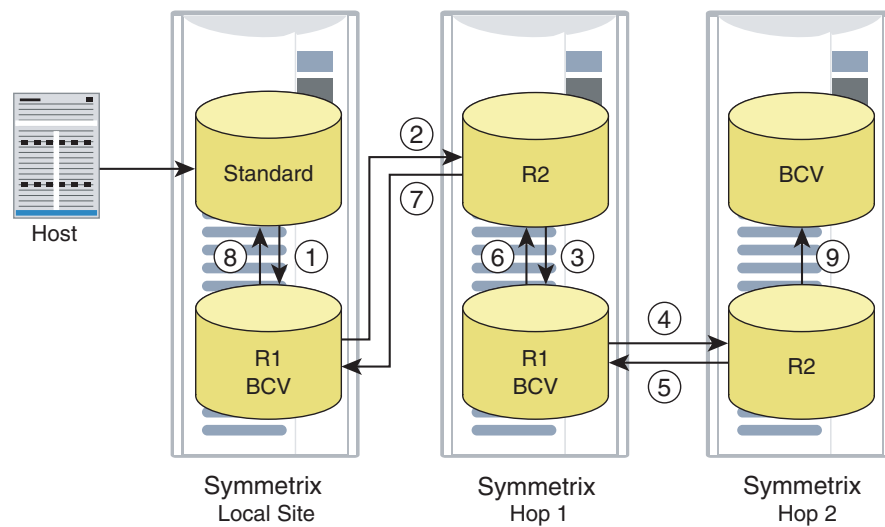
```
symmir -g Rdf1Grp restore -rdf
```

6. The `symrdf restore` command restores a data copy from the target R2 side to the source R1 side. You can initiate this operation while the BCV pair at Hop 1 is still synchronizing as a result of the action in step 5 (that is, the BCV pair state is RestInProg).

```
symrdf -g Rdf1Grp restore
```

5.4.2 Targeting Remote Devices when Remote Copying Is from a Local R1 BCV

Once a local standard device has copied data to a local RDF BCV, the BCV can be used to copy that data to remote sites. In Figure 5-4 a local R1 BCV is configured as the data source in a one-to-one relationship with a remote R2 mirror. The following commands issued from the local host show the use of various options to copy data across RDF links to targets associated with the local R1 BCV. A Regular type device group named RegGrp has been created and the appropriate devices added to it. The BCV on Hop 1 is remotely associated with the device group using `symbcv -g RegGrp -rdf -bcv` (refer to the section 5.2 *Establishing a BCV Pair with an RDF BCV* on page 5-3).



CLI-000092

Figure 5-4 Copying Data to Remote Sites from a Local R1 BCV

1. The `symmir establish` command copies data from the local standard device to the local RDF BCV (a BCV with a remote R2 mirror), forming a BCV pair. The `symmir split` command splits the BCV pair.


```
symmir -g RegGrp establish
symmir -g RegGrp verify -i 30
symmir -g RegGrp split
```
2. When established as a BCV pair, the local R1 BCV is suspended from copying data to its remote (R2) target. To resume mirroring between the two, you can split the BCV pair (as in step 1) and re-establish the SRDF pair. The `symrdf establish` command copies data from the local R1 BCV to its remote R2 mirror. The `-bcv` option signifies that the local BCV should be the source. The `symrdf split` command then splits the local BCV from its remote R2 mirror.


```
symrdf -g RegGrp establish -bcv
symrdf -g RegGrp verify -bcv -i 30
symrdf -g RegGrp split -bcv
```
3. The `symmir establish` command copies data from the Hop 1 R2 mirror to the Hop 1 BCV. The `-rdf` and `-bcv` options together in the command target the BCV that is remotely associated via the local RDF BCV device. The `symmir split` command splits the Hop 1 BCV pair.


```
symmir -g RegGrp establish -rdf -bcv
symmir -g RegGrp verify -rdf -bcv -i 30
symmir -g RegGrp split -rdf -bcv
```
4. When established as a BCV pair, the Hop 1 BCV is suspended from copying data to its remote R2 target. To resume mirroring between the two, split the BCV pair (as was done in step 3) and establish the Hop 1 BCV with its remote R2 mirror, using the `-brbcv` option to target the Hop 2 device.

The `symrdf split` command splits that SRDF pair. The Hop 2 copy is now accessible to its host.

```
symrdf -g RegGrp establish -brbcv
symrdf -g RegGrp verify -brbcv -i 30
symrdf -g RegGrp split -brbcv
```

5. If you need to restore data from the Hop 2 copy, the `symrdf restore` command restores the Hop 1 BCV from its Hop 2 remote mirror. The `symrdf split` command splits the SRDF pair.

```
symrdf -g RegGrp restore -brbcv
symrdf -g RegGrp verify -brbcv -i 30
symrdf -g RegGrp split -brbcv
```

6. The `symmir restore` command restores the Hop 1 mirror from the Hop 1 BCV copy.

```
symmir -g RegGrp restore -rdf -bcv
```

7. The `symrdf restore` command restores the local BCV from its Hop 1 mirror.

```
symrdf -g RegGrp restore -bcv
```

8. The `symmir restore` command restores the local standard device from the local BCV copy, reestablishing the local BCV pair and suspending the propagation of data from the local BCV to its remote R2 mirror. You can initiate this operation while the SRDF pair is still synchronizing as a result of the action in step 7 (that is, the SRDF pair state is `SyncInProg`).

```
symmir -g RegGrp restore
```

9. The `symmir establish` command creates a BCV copy at the Hop 2 site. The `-f` option applies a text file¹, `ProdFile`, that lists each BCV pair on a separate line with the R2 device name first and the BCV device name second (for example, 045 09C). The `-sid` option applies the command to a specified remote Symmetrix unit (`sid 056`, for example).

```
symmir -f ProdFile -sid 056 establish
```

1. Beginning with Solutions Enabler version 5.3, you can control the Hop 2 BCVs via the device group using the `-rrbcv` option. For example, `symmir -g RegGrp establish -rrbcv`. However, to do so, you need to supply the remote RDF (RA) group number (from Hop 2) to the `symbcv associate` command if you associate the Hop 2 BCVs with the device group prior to associating the Hop1 BCVs with the device group.

5.5 Using Symmetrix Automated Replication

The `symreplicate` command propagates automated incremental data copies in predefined cycles from a local Symmetrix site to a remote Symmetrix site. When used with the TimeFinder consistent split option, SRDF/AR ensures that a remotely-associated BCV always contains a DBMS restartable copy of the data.

After performing initial setup tasks, you initiate a replicate session to copy data to your target Symmetrix system according to the parameters that you define for the operation (for example, automatic copying every hour). Automatic copying continues until the session completes the number of cycles that you predefine or until you use the `symreplicate` command to stop the session. You can start, stop, restart, or query a replicate session without altering the source data copy.

You can use Symmetrix Automated Replication in a single- or multihop configuration.

5.5.1 Single-Hop Configuration

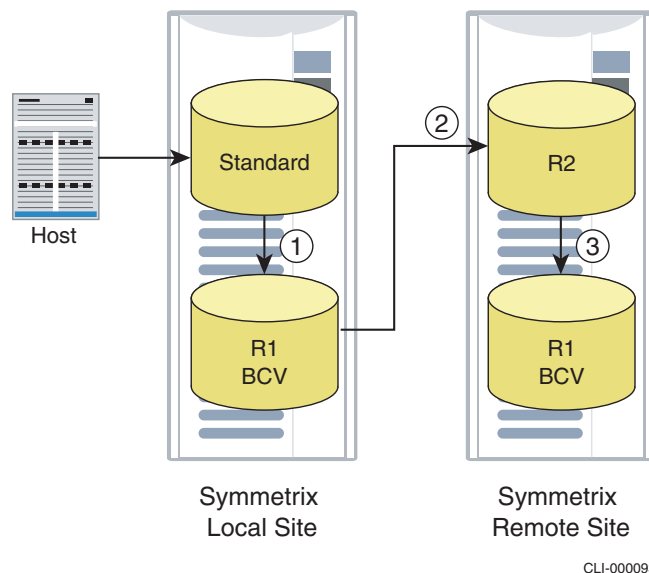


Figure 5-5 Automated Data Copying in a Single-Hop Configuration

The copy path for a single-hop configuration is from the local BCV pair (1) to the SRDF pair (2) to the remote BCV pair (3). The remotely-associated BCV holds the DBMS restartable copy.

The single-hop restart solution is applicable when you can tolerate some amount of data loss if there is a disaster at the local source site but you want a DBMS restartable copy of the data at the remote restart site.

The amount of data loss is a function of the replicate copy cycle time (period of time between the start of one copy cycle and the start of another copy cycle). Copy cycle time is affected by distance, bandwidth, I/O update rate, and locality of reference for the updates. Update rate and locality of reference tend to equate to changed tracks. The maximum data loss would be one copy cycle.

This restart solution is useful if you need to replicate long distances. The single-hop solution is also the least expensive of the restart solutions in that the single-hop solution can work

with an SRDF/DM (Data Mobility) license as well as a regular SRDF license. Moreover, this solution requires the least number of data copies.

5.5.2 Multihop Configuration

The copy path for a multihop configuration is from the local SRDF pair (1) to the remote BCV pair (2) to the remote SRDF pair (3) to the Hop 2 BCV (4). If your configuration does not include Hop 2 BCVs, the path stops at (3).

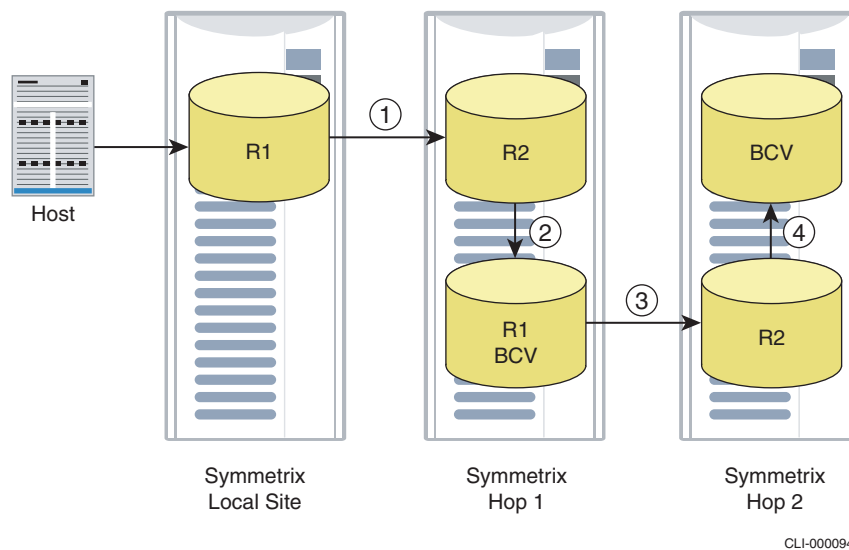


Figure 5-6 Automated Data Copying in a Multihop Configuration

The multihop restart solution is applicable when you want zero data loss in the event of a disaster at the local site. Zero data loss means that the state of the data at the Hop 2 restart site (after being propagated from the Hop 1 bunker site) is the same as it is at the local source site at the beginning of a rolling disaster.

5.5.2.1 Multihop Configuration without the Hop 2 BCV

Automated data replication in a multihop configuration *without* the BCVs at Hop 2 is applicable for those site configurations where there is little risk of losing both the local source site and the Hop 1 bunker site at the same time. If both sites were lost at the same time, the zero data loss solution is not guaranteed because I/O from the source R1 devices is being held during the consistent split of the Hop 1 BCVs and R2 devices. Depending on where in the copy cycle the failure occurred, you may not have a DBMS restartable copy of the data at the Hop 2 site.

5.5.2.2 Multihop Configuration with the Hop 2 BCV

Automated replication *with* the BCVs at Hop 2 is applicable if you want a zero data loss solution but cannot risk the loss of both the local source site and Hop 1 bunker site at the same time. With this configuration, there are two possible disaster restart possibilities:

- ◆ If only the local source site is lost, the result is zero data loss at the Hop 2 restart site.
- ◆ If both the local source site and the Hop 1 bunker site are lost, the result is a DBMS restartable copy at the Hop 2 restart site with controlled data loss. The amount of data loss will be a function of the replicate copy cycle time between the Hop 1 bunker site and the Hop 2 restart site.

5.5.3 Initial Setup Tasks for Using Symmetrix Automated Replication

The following sections describe the initial setup tasks required to establish automated replication of data:

1. Build a device group that includes devices to match one of the two configurations shown.
2. Create a replicate options file that specifies your configuration (single-hop or multihop) and defines the copy cycle parameters for your replicate session.
3. Check that the pair states are correct for a replicate session. If not, set up the correct pair states.
4. Initiate the replicate session using the `symreplicate` command and specifying the name of the replicate options file that holds your configuration and copy cycle definitions.

5.5.3.1 Building a Device Group for a Replicate Session

To build a device group for the *single-hop configuration*:

1. Display devices on the local Symmetrix and select those standard devices that you want to copy.
The standard devices should be the same type: Regular (non-RDF) or RDF1 (source). In the single-hop illustration, the device type of the local standard device is Regular.
2. Create a device group of the same type as the standard devices that you selected.
3. Add the standard devices to your device group.
4. Associate with the device group an equal number of local R1 BCV devices that match the size of the standard devices.
5. Include in the device group an equal number of remotely-associated BCVs that will hold your copies, also of matching sizes. For example:

```
symbcv -g RegularGrp associate dev 210 -rdf -bcv
```

To build a device group for the *multihop configuration*:

1. Create an RDF1 type device group.
2. Add any number of R1 standard devices to your device group (the devices that you want to copy).
3. Include in the device group an equal number of remotely-associated R1 BCV devices from the Hop 1 site that match the size of the standard devices. For example:

```
symbcv -g Rdf1Grp associate dev 1A0 -rdf
```

4. If you are using Solutions Enabler Version 5.2 or earlier, BCV devices located on Hop 2 cannot belong to the device group defined on the local host. To be able to establish the Hop 2 BCV pairs from the local host, create a device (text) file that lists the BCV pairs on Hop 2. This file must list each BCV pair with the R2 device name first and BCV device name second. For example, a file called `myfile.txt` might list two BCV pairs, 045/09C and 046/09D:

```
045 09C
046 09D
```

BCVs 09C and 09D will hold your copies unless you define the option that excludes Hop 2 BCVs. If you do not use Hop 2 BCVs, omit this step; the R2 mirrors on Hop 2 will hold your copies.

- Beginning with Solutions Enabler Version 5.3, you can omit step 4 and include the Hop 2 BCVs in the *device group* using the `-rrdf` option. (If performed before step 3, this step must include the remote RDF (RA) group number. For example, `-remote_RDFG 2`):

```
symbcv -g Rdf1Grp associate dev 09C -rrdf
symbcv -g Rdf1Grp associate dev 09D -rrdf
```

5.5.3.2 Defining Copy Cycle Parameters for a Replicate Session

You need to create a replicate options file (a text file) that specifies your replicate configuration and defines the copy cycle parameters for your replicate session. This file should conform to the following syntax, where you enter a value for the italicized text:

```
#Comment
SYMCLI_REPLICATE_HOP_TYPE=ReplicateConfig
SYMCLI_REPLICATE_USE_FINAL_BCV=TrueOrFalse
SYMCLI_REPLICATE_CYCLE=CycleTime
SYMCLI_REPLICATE_NUM_CYCLES=NumberOfCycles
SYMCLI_REPLICATE_CYCLE_OVERFLOW=OverflowBehavior
SYMCLI_REPLICATE_CYCLE_DELAY=DelayTime
```

The *ReplicateConfig* value must be either `SINGLE` or `MULTI` for a single-hop or multihop configuration, respectively.

The *TrueOrFalse* value specifies whether the Hop 2 Symmetrix has BCVs that participate in the multihop configuration. The default is `TRUE`. If your `MULTI` configuration does not use BCVs at Hop 2, set the value to `FALSE`.

The *CycleTime* value specifies the period of time between the start of one copy cycle and the start of another copy cycle, provided that you have specified more than one cycle. You can express this value in *minutes* or *hours:minutes* (*hh:mm*). For example, a value of 60 would start a new cycle every hour.

The *NumberOfCycles* value specifies the number of cycles to perform before ending the replicate session. A zero value (or not setting this parameter) results in continuous cycling until you specifically stop the replicate session.

The *OverflowBehavior* value specifies what to do if the copy cycle runs longer than the time you specified for the cycle to complete. You can specify `NEXT` to skip a cycle and wait for the next cycle to begin, or specify `IMMEDIATE` to begin the next cycle immediately. For example, if you specified a one-hour copy cycle and the copy took 1.5 hours to complete, the next cycle could begin in half an hour or immediately.

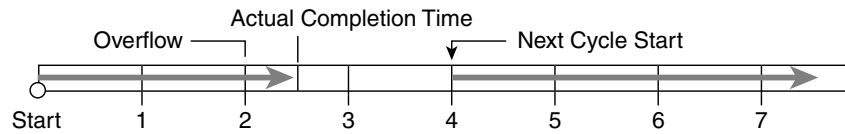
The *DelayTime* value specifies the minimum amount of time to wait between the end of one cycle and the beginning of the next cycle. Even if a cycle runs longer than the specified cycle time and the overflow behavior is set to immediate, the system waits for the duration of the *DelayTime* value before beginning another cycle. For example, a value of 20 provides at least a 20-minute delay between cycles.

If you are unsure about choosing appropriate cycle time parameters the first time, EMC recommends starting with a loose time constraint and tightening parameters when you have a better idea of data size and SRDF throughput expectations. You could begin with either of the following:

- ◆ *CycleTime=60, DelayTime=0, OverflowBehavior=NEXT*
New cycle every hour, if possible; otherwise, every two or three hours based on data throughput and size.
- ◆ *CycleTime=0, DelayTime=60*
Complete the first copy cycle, then wait an hour; then another cycle and a one-hour wait; and so forth.

Figure 5-7 shows different combinations of cycle time, delay time, and overflow behavior to achieve various results. Reference points on the cycle time lines are marked ACT (actual completion time), NCS (next cycle start), and MDT (minimum delay time). Short cycle and delay times (in minutes) were chosen for illustration purposes only.

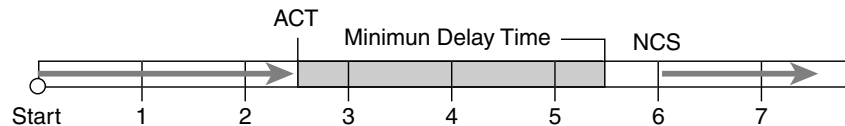
1. Cycle Time = 2, Delay Time = 0, Overflow = Next



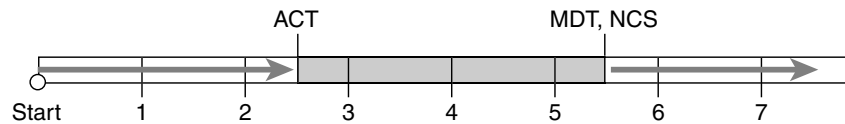
2. Cycle Time = 2, Delay Time = 0, Overflow = Immediate



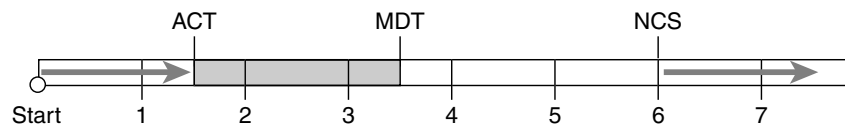
3. Cycle Time = 2, Delay Time = 3, Overflow = Next



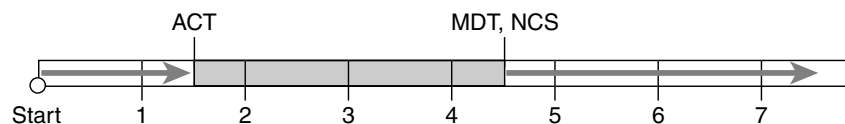
4. Cycle Time = 2, Delay Time = 3, Overflow = Immediate



5. Cycle Time = 3, Delay Time = 2, Overflow = Next



6. Cycle Time = 0, Delay Time = 3, Overflow = Next or Immediate



CLI-000061

Figure 5-7 Copy Cycle Results for Different Replicate Cycle Parameter Settings

Copy cycles #1 and #2 have the same cycle time (2) and no delay time. When copy cycle #1 runs longer than two minutes, the overflow setting of "Next" results in a new copy cycle beginning at the four-minute mark.

When copy cycle #2 runs longer than two minutes, the overflow setting of "Immediate" results in a new copy cycle beginning as soon as that copy cycle completes.

Copy cycles #3 and #4 have the same cycle time (2) and the same delay time (3). When copy cycle #3 runs longer than two minutes, the system waits three minutes and then resumes copying at the next scheduled copy cycle (the six-minute mark). When copy cycle #4 runs

longer than two minutes, the system waits three minutes and then resumes copying as soon as the wait period completes.

Copy cycle #5 finishes before its scheduled cycle time of three minutes, waits two minutes, and begins copying again at the next scheduled copy cycle (the six-minute mark).

Copy cycle #6 performs continuous copy cycles when its cycle time is set to zero. When copying completes, the system waits three minutes and then begins another copy cycle, regardless of the overflow setting.

5.5.3.3 Setting Up Pair States for a Replicate Session

To begin a replicate session in the *single-hop configuration*, the local BCV pair must be synchronized, the SRDF pair must be suspended, and the remote BCV pair must be split. To put the pairs in these states initially, carry out the following steps:

1. Perform a full establish on the local BCV pair. When the establish operation completes, perform a TimeFinder split.
2. Perform an incremental establish on the SRDF pair.
3. Perform a full establish on the remote BCV pair. When the establish operation completes, perform a TimeFinder split.
4. Reestablish the local BCV pair. A side effect of this operation is that the SRDF pair is suspended.

To begin a replicate session in the *multihop configuration*, the local SRDF pair must be synchronized, the BCV pair on Hop 1 must be synchronized, the remote SRDF pair must be suspended, and the BCV pair on Hop 2 must be split. To put the pairs in these states initially, carry out the following steps:

1. When EMC installs an SRDF configuration, the installers usually establish SRDF pairs at that time.
If the local R1 standard device is synchronized with its remote R2 mirror, the first criterion is fulfilled. If not, then fully establish the local SRDF pair.
2. Perform a full establish on the Hop 1 BCV pair. When the establish operation completes, perform a split on the Hop 1 BCV pair.
3. A side effect of establishing the BCV pair in step 2 is the suspension of mirroring between the R1 BCV on Hop 1 and its R2 mirror on Hop 2. Perform an incremental establish now on that remote SRDF pair.
4. Perform a full establish on the Hop 2 BCV pair, specifying either a device file that lists the BCV pair or, beginning with Solutions Enabler version 5.3, the device group with the `-rrbcv` option. When the establish operation completes, perform a split on the Hop 2 BCV pair. However, if your configuration does not use Hop 2 BCVs, then skip this step.
5. Reestablish the Hop 1 BCV pair. A side effect of this operation is that the remote SRDF pair is suspended.

5.5.3.4 Starting and Stopping a Replicate Session

The `symreplicate` command allows you start, stop, or restart a replicate session. By default, the replicate session exists as a background process. However, both `symreplicate start` with the `-foreground` option and `symreplicate query` display status of a replicate session.

Device locks are held during the replicate session to block other applications from altering device states while this session executes. Any abnormal termination of the replicate session may require you to restart the session by first releasing the locks manually or, beginning with Solutions Enabler Version 5.2, by recovering the existing device locks using the `-recover` option (see Appendix B, Example 6).

A replicate session normally ends when it completes the number of cycles that you specified in your replicate options file or when you issue a `symreplicate stop` command. If you stop the session explicitly, you can use the `symreplicate restart` command to restart the replicate session at the spot where it stopped.

The following `symreplicate start` example specifies a replicate options file named `opfile.txt`. Assume that `opfile.txt` contains the following lines:

```
#Copy the data in a single-hop configuration...
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
#Every 15 minutes...
SYMCLI_REPLICATE_CYCLE=15
#Or as often as possible if 15 minutes is not enough
SYMCLI_REPLICATE_CYCLE_OVERFLOW=IMMEDIATE
```

Assuming that all the initial setup tasks are complete, the example starts a replicate session for a device group named `RegularGrp`. The `symreplicate start` command initiates the first replicate copy cycle. Subsequent commands query the replicate session, stop it, and restart it.

```
symreplicate -g RegularGrp -options opfile.txt start
symreplicate -g RegularGrp query
symreplicate -g RegularGrp stop
symreplicate -g RegularGrp restart
```

These `symreplicate` commands allow you to include options on the command line that are described in *Solutions Enabler Symmetrix SRDF CLI Product Guide*.

5.5.3.5 Restarting a Replicate Session in a Clustered SRDF/AR Environment

Beginning with Solutions Enabler Version 5.3 and Enginuity Version 5669, you can instruct the replicate session to write its log file to a locally attached Symmetrix unit instead of to a host-based file. This capability allows you to restart the replicate session from another node in a clustered SRDF/AR environment if the primary node drops.

To enable this capability, you need to write the SRDF/AR log file to the Symmetrix File System (SFS). SRDF/AR supports an SFS restart on any host that is locally attached to the Symmetrix unit that contains the log file. Thus, the Symmetrix to which you write the log file must be part of the device group and connected locally to a host in the cluster.

You write the SRDF/AR log file to SFS by supplying the ID of the Symmetrix that will contain the log file. Optionally, you can also specify the `-log` option and a file name (such as `log2`). By default, SRDF/AR uses the device group name as the name of the log file.

```
symreplicate -g symrep start -sid 202 -log log2
```

If you do specify a log file name when starting a replicate session, you must also specify that same log file (`-log LogFileName` instead of `-g DeviceGroupName`) when issuing `symreplicate stop`, `restart`, `query`, `show`, and `delete` commands for the same replicate session.

If the host that is running the replicate session fails, you can restart the session from another host that is locally attached to Symmetrix 202. For example, the `symreplicate restart` command below uses both the `-sid` option (to specify where the SRDF/AR log file is located) and the `-log` option (to specify the log file name that was used to start the replicate session):

```
symreplicate -log log2 restart -sid 202 -recover
```

SRDF/AR uses information from the log file to restart. Therefore, the device group does not need to be defined on the restart host. The `-recover` option tells SRDF/AR to attempt to recover the device locks.

The `symreplicate list` command with the `-sid` option allows you to display a list of SRDF/AR log files that exist in the SFS on the specified Symmetrix unit. The following output indicates that the log file `symrep` was generated from the device group name. (SRDF/AR uppercases all log file names that are generated by `symreplicate start` when stored in SFS).

```
symreplicate list -sid 202
```

Log file listing for Symmetrix ID 000000006202

Name	Log Type
LOG1	User-specified
LOG2	User-specified
SYMREP	Device group

The `symreplicate delete` command with the `-sid` option and the `-log` option or `-g` option allows you to delete a log file from the SFS on a Symmetrix unit. For example:

```
symreplicate delete -log log2 -sid 202
symreplicate delete -g symrep -sid 202
```

Keep in mind that the `-sid` parameter must be used to create an SRDF/AR log file in the SFS that permits restarting a replicate session from a host other than the primary host.

5.5.3.6 Error Handling for a Replicate Session

You can include entries in your replicate options file that allow you to log a session and set time limits for retrying operations when errors occur or when no data is flowing. Your entries in the replicate options file should conform to the following syntax, where you enter a value for the italicized text:

```
SYMCLI_REPLICATE_LOG_STEP=TrueOrFalse
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
```

The *TrueOrFalse* value specifies whether steps in a replicate session are logged to the SYMAPI log file after each step completes. If the parameter is set to `TRUE`, log file entries show the time each step ended and whether the step was successful.

The *TimeLimit* value controls SRDF/AR retry operations when an error occurs continuously or when no data has flowed for the duration of the timer. The general time limit (GEN) parameter controls how long to retry recoverable errors that occur when either a BCV or RDF control operation fails (for example, failing to acquire a Symmetrix lock). The BCV and RDF time limits control how long to retry errors (or wait for data to flow) when BCV or RDF devices are trying to reach a certain state.

You can specify the *TimeLimit* value in hours-and-minutes (HH:MM) or in seconds (SSS). The value zero (0) causes the operation to be retried indefinitely. Default time limits in hours-and-minutes are:

```
SYMCLI_REPLICATE_GEN_TIME_LIMIT=00:30
SYMCLI_REPLICATE_RDF_TIME_LIMIT=02:00
SYMCLI_REPLICATE_BCV_TIME_LIMIT=04:00
```

The *SleepTime* value specifies the *minimum* time that SRDF/AR should sleep before rechecking if devices have entered a specific state or retrying an operation as a result of a recoverable error. During normal step processing, SRDF/AR determines that a step operation is complete when devices have entered a specific state. If the step operation is not complete, SRDF/AR sleeps for a period of time, checks device states again, and then calculates sleep time based on the number of invalid tracks and the rate of data movement. Therefore, SRDF/AR's actual sleeping time may be greater than the *SleepTime* value. The default values for GEN, BCV, and RDF are 10, 10, and 15 seconds, respectively.

The *Factor* value multiplied by the *SleepTime* value specifies the *maximum* time that SRDF/AR sleeps before rechecking if devices have entered a specific state. By default, when checking the state of BCV devices, SRDF/AR sleeps between 10 and 30 seconds up to a maximum of two hours. When checking RDF devices, sleep time is between 15 and 60 seconds up to a maximum of four hours.

Running a Step Error Script File during a Replicate Session

The `symreplicate start` command includes a script file option (`-steperror ScriptFileName`) that can perform file-specified actions if SRDF/AR encounters an error during the replicate session. Diagnostic information about the error is provided through various command line arguments (refer to *Solutions Enabler Symmetrix SRDF CLI Product Guide*).

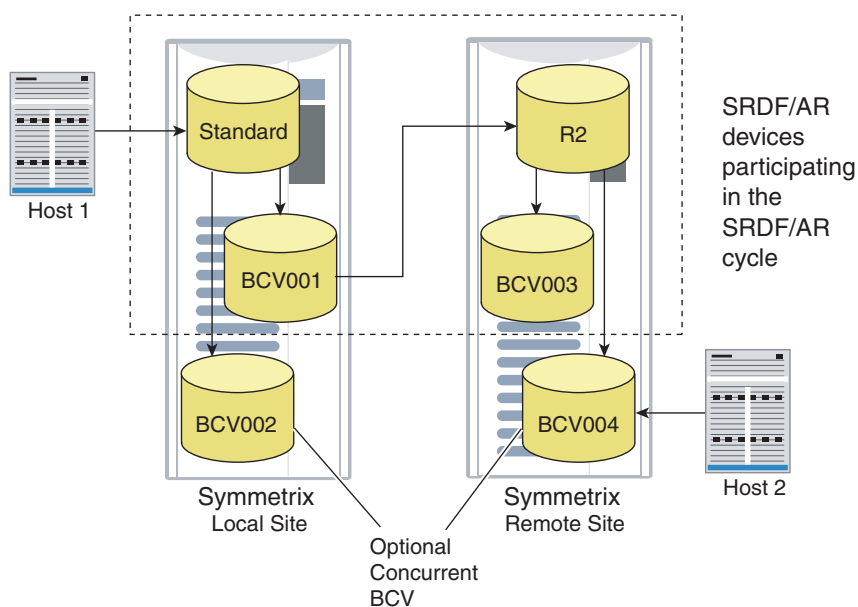
Specifying this option does not replace the need to check for non-zero exit status from SRDF/AR. If an error occurs when a replicate session tries to start (causing the non-zero exit status), then the `steperror` script file will not execute. The `-steperror` option is not available on all platforms.

5.6 Accessing a Concurrent BCV while Running SRDF/AR

If you need to have an independent copy of the data during a SRDF/AR cycle, you can do so by setting up concurrent BCVs such that one BCV is included in the SRDF/AR devices and the other BCV is not included in the SRDF/AR devices. The BCV that is not part of the SRDF/AR cycle receives the same data as a BCV that is part of the SRDF/AR devices. To set this up, include the non-SRDF/AR BCV and its paired SRDF/AR standard device in a device file (or in a device group on a different host¹ from the one that defines the SRDF/AR device group). Then establish the non-SRDF/AR BCV with the SRDF/AR standard device so that the standard copies data to both BCVs — the SRDF/AR BCV and non-SRDF/AR BCV.

Participating devices are locked during an SRDF/AR copy cycle. A `symmir` option (`-skip`) allows you to override that lock on the standard devices in order to establish a non-SRDF/AR BCV concurrently with a standard device that is also established with an SRDF/AR BCV during the SRDF/AR copy cycle.

In Figure 5-8, BCV001 and BCV002 are concurrent BCVs, as are BCV003 and BCV004. But BCV002 and BCV004 are not among the devices participating in the SRDF/AR copy cycle. By defining a device file on Host 1 that includes the standard device and BCV002, you can establish this BCV pair during the copy cycle, split the pair, and then access the data on BCV002 without interfering with the copy cycle.



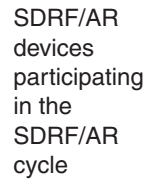
CLI-000076

Figure 5-8 Accessing Concurrent BCVs in a Single-Hop Configuration

You can also access BCV004 during the SRDF/AR copy cycle. By defining a device group (or device file) on Host 2 that includes the R2 and BCV004, you can establish that BCV pair, split it, and access BCV004.

Figure 5-9 illustrates a multihop configuration with concurrent BCVs at the Hop 1 and Hop 2 sites.

1. A device can be included in only one device group on the same host.



CLI-000077

Figure 5-9 Accessing a Concurrent BGV in a Multihop Configuration

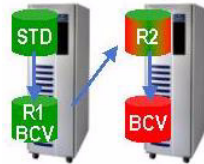
This chapter provides the following information:

- ◆ 6.1 Overview.....6-2
- ◆ 6.2 Prior to the Disaster.....6-3
- ◆ 6.3 Disaster Restart Procedures for a Single-Hop Configuration6-5
- ◆ 6.4 Disaster Restart Procedures for a Multihop Configuration.....6-13

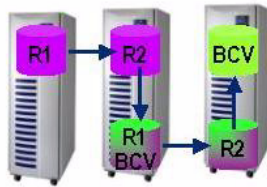
6.1 Overview

Although application testing procedures and actual disaster restart command procedures must be customized according to the needs of each customer, this chapter will discuss generic disaster restart procedures. The following procedures will be covered in this chapter:

Single-hop disaster restart procedures



Multihop disaster restart procedures



The reason to restart at the remote site could vary. There may be an actual disaster either by losing the entire site, host-only impact, or any other unplanned event that forces the restart of the applications at the remote site. These are the procedures used in these types of unplanned outages, sometimes referred to as a disaster.

6.2 Prior to the Disaster

The following tasks need to be performed before the disaster:

- ◆ Define and document corresponding device group or device files for use at the target site.
- ◆ Define application-testing procedures versus actual disaster restart command procedures.
- ◆ Perform application-testing procedures.
- ◆ Test actual disaster restart commands procedures.
- ◆ Configure hardware and software for testing and restart.

In the event of an unplanned outage, EMC recommends the restart off the R2 devices. The main reason for this recommendation is because the environment is better prepared for the return home procedures discussed in Chapter 7. Other reasons to restart off the R2 devices is to maintain a point-of-disaster image on the set of BCVs, or to use these BCVs for other purposes while the production is running at the remote site. The decision of what devices to use at the remote site will aid in the decision of which devices to protect. This in turn will aid in the decision of how to use the BCVs in the event of a disaster.

If a decision is made to restart off the BCVs, it is important to provide RAID protection to the BCVs. Note however, if the BCVs are RAID protected, the SRDF/AR cycle could be elongated.

6.2.1 Corresponding Device Groups or Device Files on Target Host

Once a disaster occurs, it will be necessary to perform queries to determine which set of devices, R2s or BCVs, contain the most current copy of data. In order to optimize the RTO, device groups or device files can be created on a target host prior to the disaster. These groups or files should be used for the application testing procedures and the actual disaster restart command procedure test. If a target host is not available, the device files could be set up on the source host, and included in the replication process. It is recommended that a separate logical device be used to contain such documentation, i.e., not a database device.

Single-hop configurations will require: device file for source STD and R1-BCV pairs, device file for source R1-BCV and target R2 pairs, and device file for target R2 and BCV pairs. Multihop configurations will require: device file for source R1 and bunker R2 pairs, device file for bunker R2 and R1-BCV pairs, device file for bunker R1-BCV and target R2 pairs, and device file for target R2 and BCV pairs.

6.2.2 Application Testing Procedures versus Actual Disaster Restart Command Procedures

The application testing procedures may vary slightly from the procedures used during the actual disaster or unplanned outage. During the application testing procedures, the customer may want to continue to process Symmetrix Automation Replication cycles to protect from a true disaster or unplanned outage. In this situation, a concurrent BCV would be used to perform the application testing procedures. Because different devices are used for the application testing, another set of procedures is required and should be tested. These procedures will be referred to as actual disaster restart command procedures.

6.2.3 Application Testing Procedures

It is recommended to perform routine testing to bring up the operating systems, mount the file systems, and restart the databases and applications. These procedures could be tested from a separate set of devices, such as concurrent BCVs in order to continue processing the Symmetrix Automated Replication cycles. This will provide protection from a disaster during the test. Although the devices will be different during a true disaster, this test will prepare the applications and database personnel for such an event.

6.2.4 Actual Disaster Restart Command Procedures

It is recommended to test the procedures that will be used during a true disaster or unplanned outage. This will verify the device configuration and commands used during such an event. Terminating the Symmetrix Automated Replication process or pulling the SRDF links will simulate a disaster. The devices files mentioned above, as well as the necessary queries, will be used to determine your most current copy of data.

All testing procedures (application or command) should be well documented and tested prior to a disaster or unplanned outage.

6.2.5 Hardware and Software Configuration at the Target Site

The procedures to determine the most current copy of data can be executed by a Customer Engineer using the service processor. To automate this procedure, it is necessary to have a UNIX or Windows Server connected to the target Symmetrix system. Solutions Enabler, along with a TimeFinder and SRDF license, will need to be installed. The purpose of this server is to send commands to the storage system. Therefore, during testing or in the event of a disaster, EMC recommends the server be brought up in single-user mode. This will prevent unwanted updates or access to the data until the most current copy is verified. It is recommended to have devices groups or device files predefined for the procedures.

6.3 Disaster Restart Procedures for a Single-Hop Configuration

Figure 6-1 is an overview picture of a disaster that has occurred for a single-hop configuration.

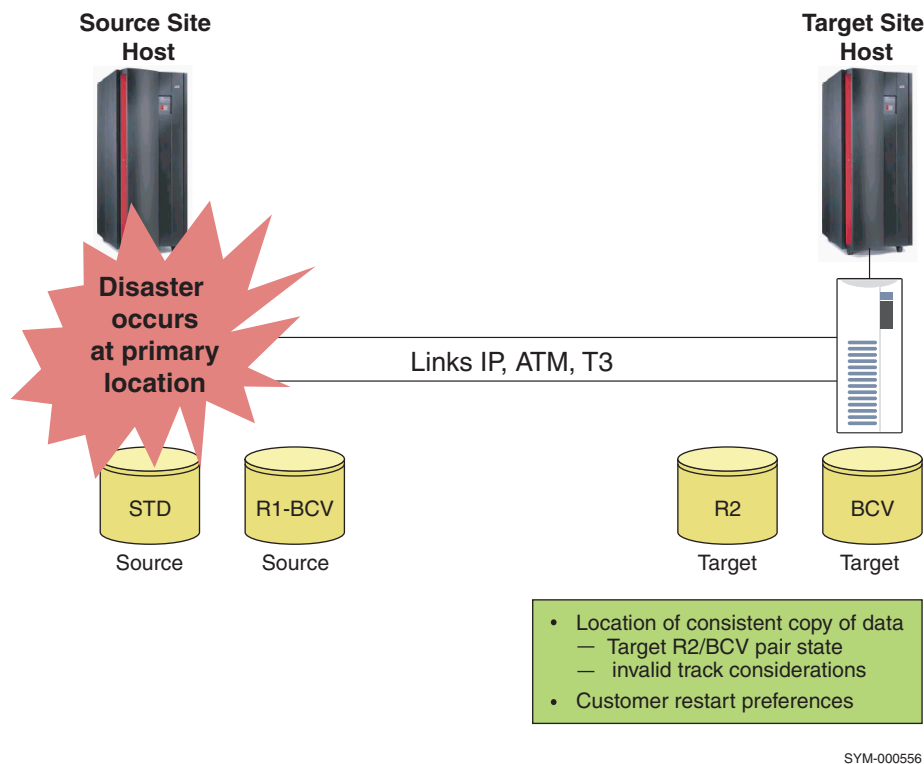


Figure 6-1 Single-Hop Source Disaster

When a disaster happens in the primary site, the target site will always contain a consistent and restartable copy of data on the BCV devices. However, depending on the data replication step that was being executed at the moment of the disaster, the R2s could contain the most recent consistent copy. Therefore, the first procedure to be followed is to determine the status of the target BCV with respect to the target R2 devices.

6.3.1 Determining the Most Current Consistent Copy and Restarting Applications

To determine the location of the most current copy of data, the target R2s and target BCVs will need to be queried. When issuing the queries from the target site, be sure to use the appropriate previously created device file, the Symmetrix ID, and the RDF group if querying or verifying RDF status.

Figure 6-2 is an example of the complete device list.

Source STDs			Source R1-BCVs			Target R2s	Target BCVs	
Physical Device	Symm Dev	Logical Dev	Physical Device	Symm Dev	Logical Dev	Symm Dev	Symm Dev	Logical Dev
	001			00F		039	040	
	002			010		03A	041	
	003			011		03B	042	
	004			012		03C	043	
	005			013		03D	044	
	006			014		03E	045	
	007			015		03F	046	

Figure 6-2 Complete Single-Hop Device List

Figure 6-3 is an example of the three corresponding device files that will be used in the command structures.

std 2 r1bcv		r1bcv 2 r2		r2 2 brbcv	
001	00F	00F	039	039	040
002	010	010	03A	03A	041
003	011	011	03B	03B	042
004	012	012	03C	03C	043
005	013	013	03D	03D	044
006	014	014	03E	03E	045
007	015	015	03F	03F	046

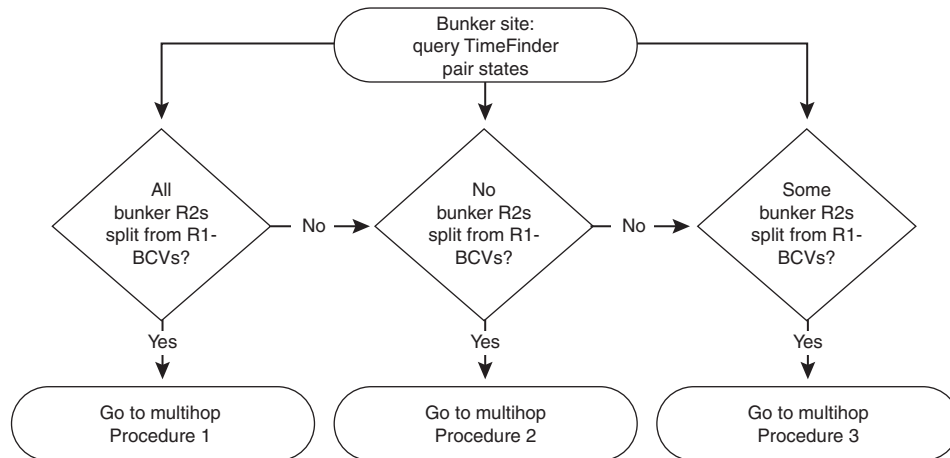
Figure 6-3 Devices Files Used for Single-Hop Restart Procedures

The following is an example command, issued from the target site, used to perform the verification of the state of the target R2s and target BCVs:

```
symmir -f <DevFile> query -sid <SymmID>
```

The nature of the Symmetrix Automated Replication cycle allows the determination of the devices housing the most current data based on the results of the query command. The flowchart in Figure 6-4 uses these results to direct us to the appropriate restart procedure. There are three possible states of the R2/BCV devices:

- ◆ **All** of the target (R2) devices are split from their partner BCV devices
- ◆ **None** of the target (R2) devices are split from their partner BCV devices
- ◆ **Some** of the target (R2) devices are split from their partner BCV devices

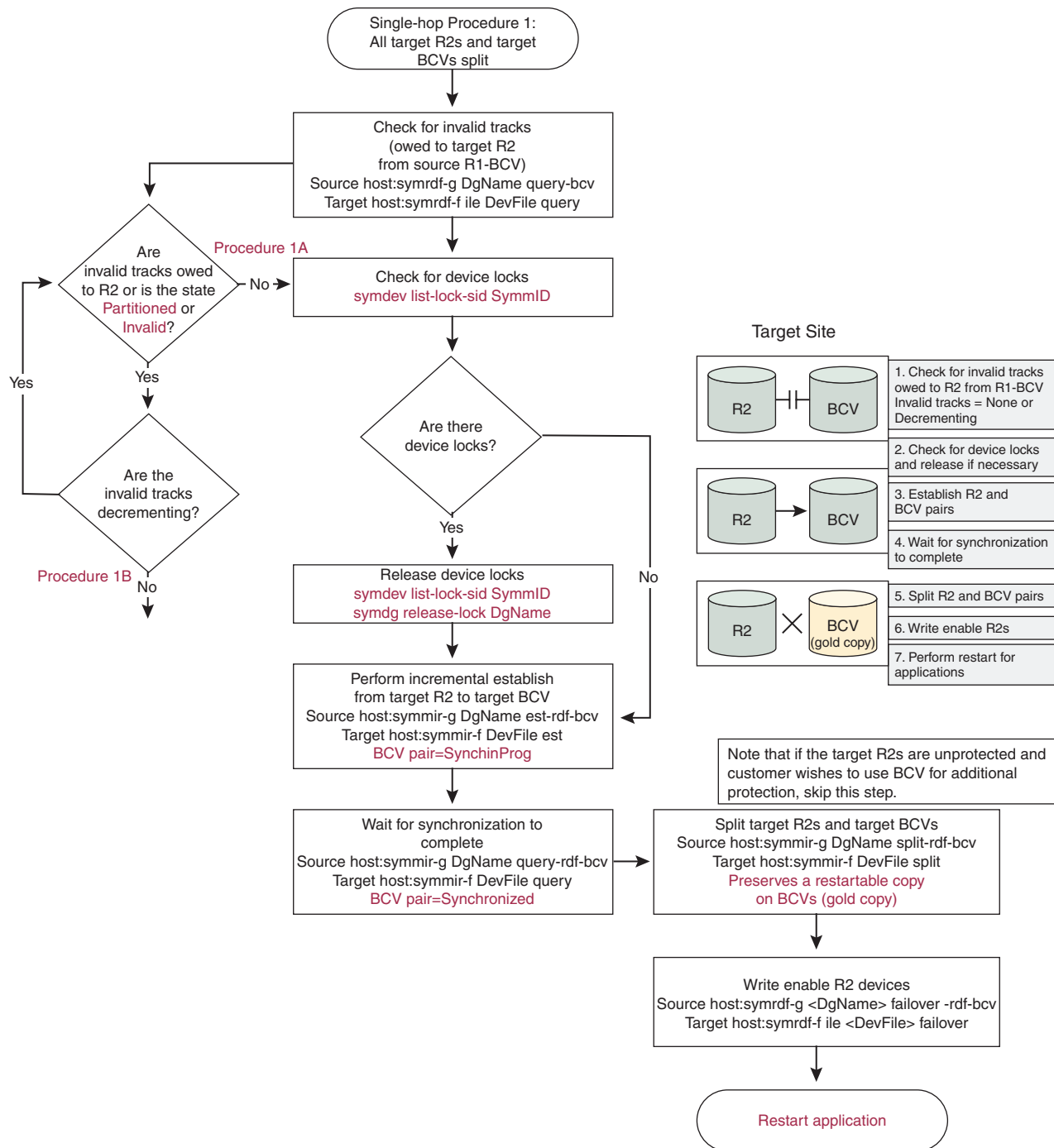


SYM-000609

Figure 6-4 Query Target TimeFinder Pair State

6.3.1.1 Procedure 1: All of the Target (R2) Devices Are Split from Their Partner BCV Devices

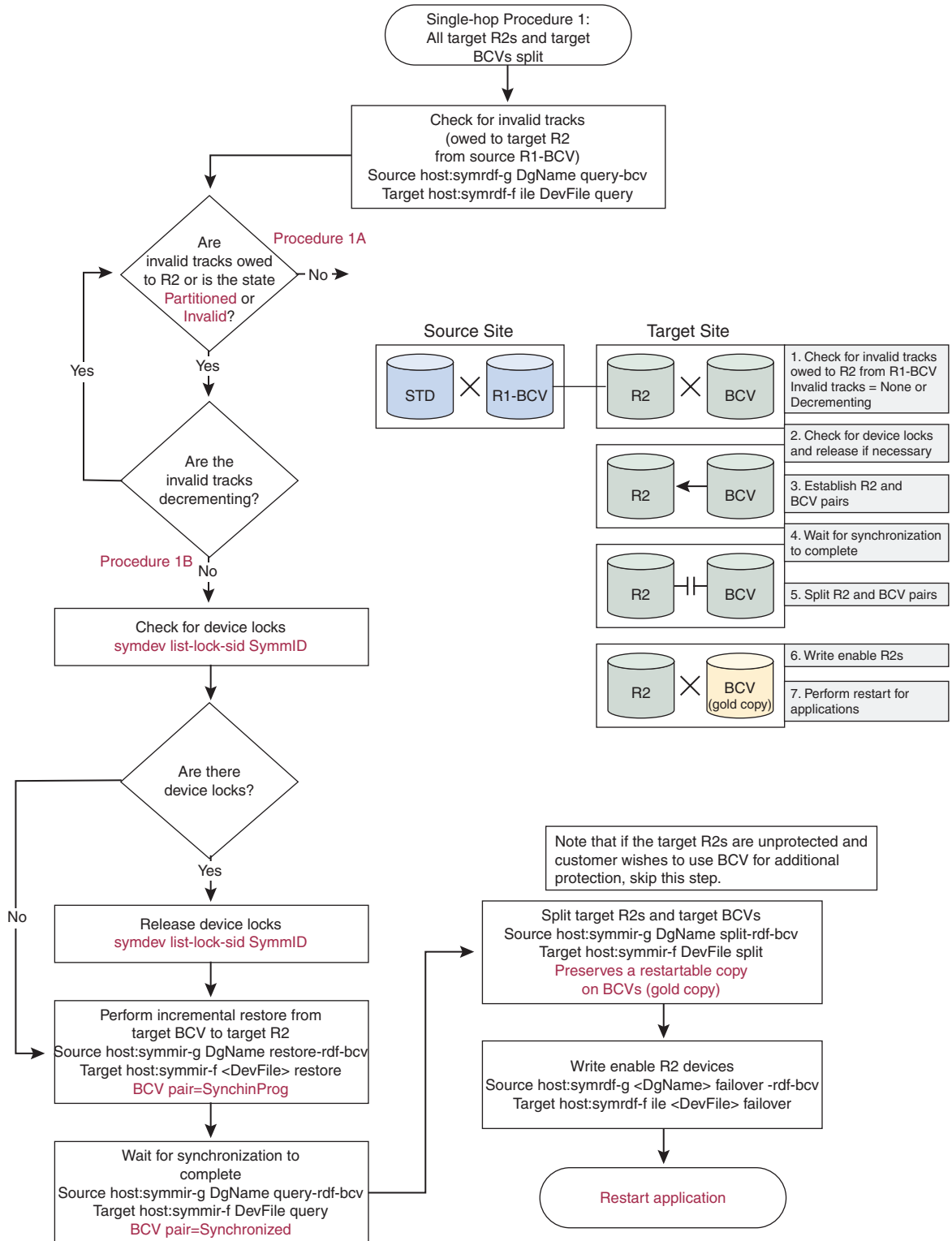
All of the target R2s and target BCVs are in a split state. This indicates that the cycle was in the process of sending the data from the source R1-BCVs to the target R2s. The target BCVs contain the most recent copy of data, unless the target R2s are still receiving data from the source R1-BCVs. An example of how this could occur is if the disaster only affected the source hosts. Figure 6-5 details the steps as if the target R2s were still receiving updates from the source R1-BCVs.



SYM-000610

Figure 6-5 Restart Procedure 1A: R2s Contain Most Recent Cycle of Data

If the R2s have invalid tracks owed from the R1-BCVs and the count is not decrementing, the R2s devices contain an incomplete copy. The flowchart in Figure 6-7 demonstrates the steps if the R2s are not receiving updates from the source site and they have invalid tracks. The most recent image will reside on the target BCVs.

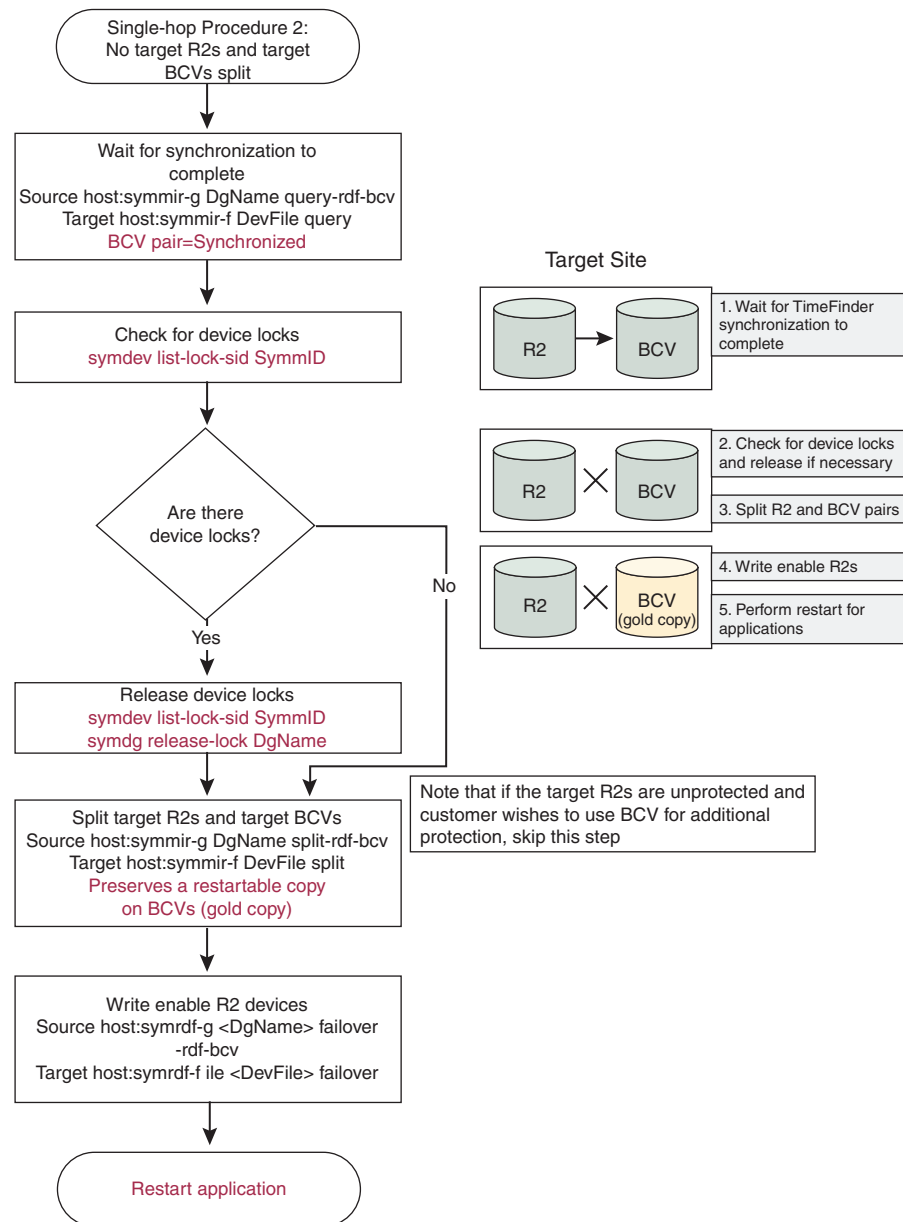


SYM-000611

Figure 6-6 Restart Procedure 1B: BCVs Contain Most Recent Cycle of Data

6.3.1.2 Procedure 2: None of the Target (R2) Devices Are Split from Their Partner BCV Devices

If all of the target R2s and BCVs are either synchronized or synchronization is in progress, we can deduce the cycle was in the process of copying the most recent copy of data to the target BCVs from the target R2s. Once this copy is complete, both the R2s and the BCVs will contain the most recent copy of data. The flowchart in Figure 6-7 details the steps for this scenario.

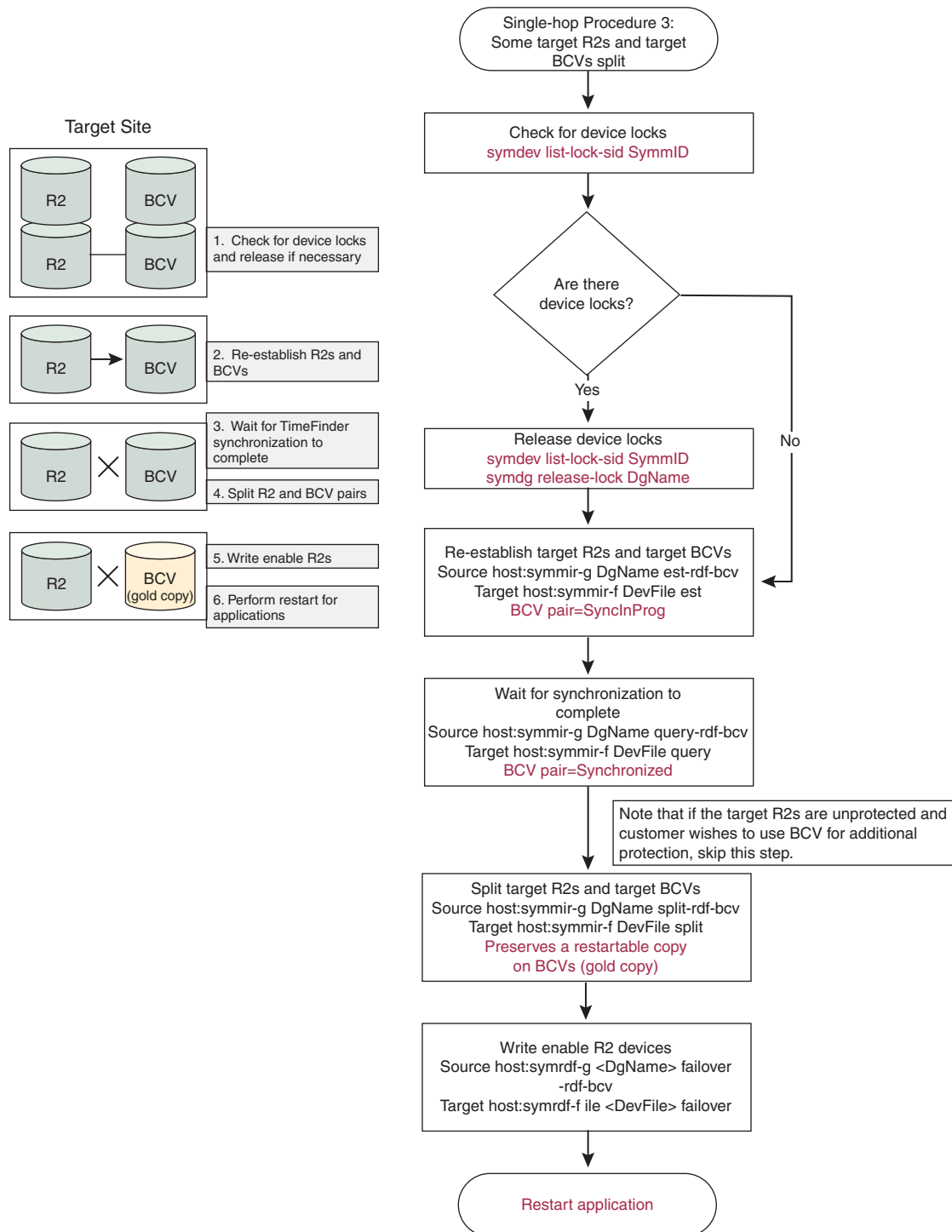


SYM-000612

Figure 6-7 Restart Procedure 2: R2s and BCVs Contain Most Recent Cycle of Data

6.3.1.3 Procedure 3: Some of the Target (R2) Devices Are Split from Their Partner BCV Devices

If the target R2s and target BCVs are in a mixed state (some synchronized and some split), the cycle was in the process of either establishing or splitting the target BCVs from their R2s. The guaranteed most recent copy of data will reside on the target R2s. The flowchart in Figure 6-8 details the steps for this scenario.



SYM-000613

Figure 6-8 Restart Procedure 3: RR2s Contain Most Recent Cycle of Data

6.3.2 Make the Devices Ready to the Host

The R2 devices now contain the most recent restartable copy of data and can be made read/write enabled to the target host. The following command accomplishes this task.

```
symrdf -f <DevFile> -sid <SymmID> -rdfg <RdgGrp> rw_enable r2
```

This command has a similar effect as a `symrdf failover` command, but does not require the SRDF links to be available. The `symrdf failover` command performs a write-disable operation on the R1 devices, suspends the logical link between the R1 and R2 device, and performs the write-enable operation for the R2 devices.

```
symrdf -f <DevFile> -sid <SymmID> -rdfg <RdgGrp> failover
```

The `symdev` command can be used to change the state of individual devices.

```
symdev <SymDevName> -sid <SymmID> rw_enable
```

6.3.3 Restarting the Applications

The environment is now ready to be restarted, which could include the following operations:

- ◆ Import the volumes groups.
- ◆ Perform `fsck` on the file systems.
- ◆ Mount the file systems.
- ◆ Bring up the databases and applications.

Please consult your operating system procedures.

6.3.4 Disaster Restart Off the BCVs

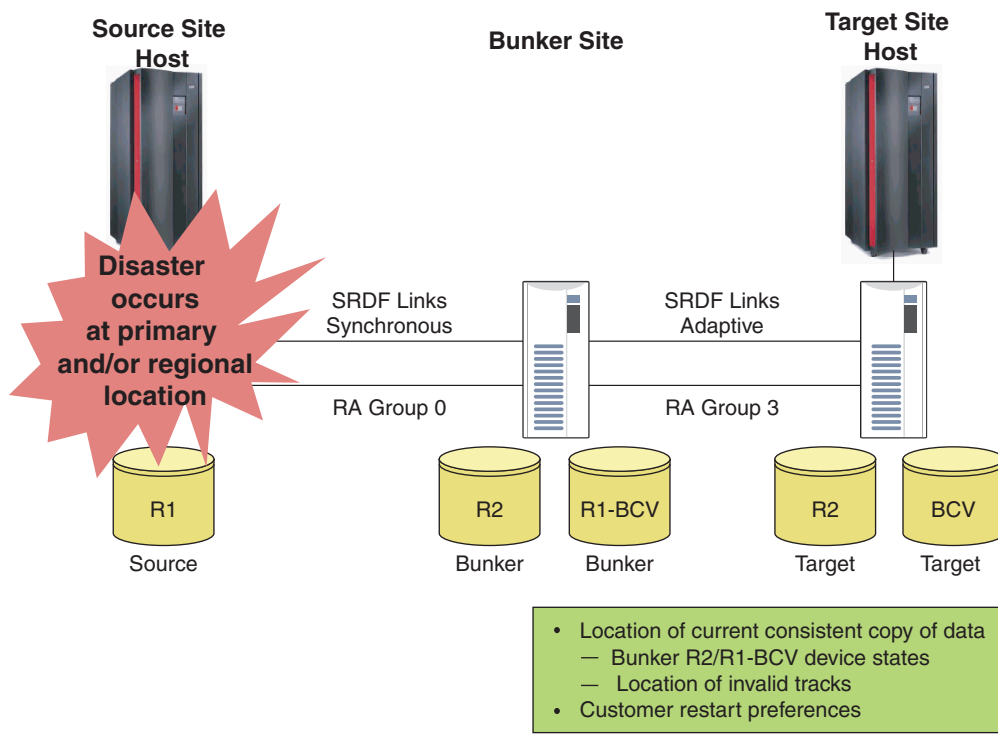
The difference in the procedure(s) to restart off the target R2 devices and the procedure(s) to restart off the target BCV devices is where the most current restartable copy is staged. If the most recent copy is on the target R2 devices, these devices will need to be established with the target BCVs, and then split before restarting the environment on the target BCVs. If the most recent copy is on the target BCVs, a restore can be done to save a point-of-disaster copy. A split is required before restarting the environment on the target BCVs.

There is no reason to read/write enable the R2 devices. The target BCV devices should be protected and addressable to the target host.

6.4 Disaster Restart Procedures for a Multihop Configuration

If a disaster occurs in the primary site, the target site will contain a consistent and restartable copy of data on the BCV devices. The bunker site will have zero data lost at the point of the beginning of the rolling disaster of the production data. This is naturally dependent write-consistent (i.e., restartable image) in a single Symmetrix source system to single Symmetrix bunker system. One more cycle of data replication from the bunker site to the target site must be executed to propagate the point-of-disaster image of data. This operation will be executed from the target site.

Figure 6-9 is an overview picture of a disaster that has occurred for a multihop configuration.



SYM-000557

Figure 6-9 Multihop Source Disaster

When a disaster happens in the primary site, the target site will always contain a consistent and restartable copy of data on the BCV devices. However, depending on the data replication step that was being executed at the moment of the disaster, the bunker R2s could contain the most recent consistent copy. Therefore, the first procedure to be followed is to determine the status of the bunker BCVs with respect to the bunker R2 devices.

6.4.1 Determining the Most Current Consistent Copy and Restarting Applications

To determine the location of the most current copy of data, the bunker R2s and bunker BCVs relationship will need to be queried. When issuing the queries from the target site, be sure to use the appropriate previously created device file, the Symmetrix ID, and the RDF group if querying or verifying RDF status.

Figure 6-11 is a list of devices files used for multihop restart procedures. Figure 6-10 is an example of the complete device list.

Source R1s			Bunker R2s	Bunker R1-BCVs		Target R2s		Target BCVs
Physical Device	Symm Dev	Logical Dev	Symm Dev	Symm Dev	Logical Dev	Symm Dev	Logical Dev	Symm Dev
	032		063	04E		028		04B
	033		064	04F		029		04C
	034		065	050		02A		04D
	035		066	051		02B		04E
	036		067	052		02C		04F
	037		068	053		02D		050
	038		069	054		02E		051

Figure 6-10 Complete Multihop Device List

Figure 6-11 is an example of the four corresponding device files that will be used in the command structures.

r1 2 r2		r2 2 r1bcv		r1bcv 2 r2		r2 2 r1bcv	
032	063	063	04E	04E	028	028	04B
033	064	064	04F	04F	029	029	04C
034	065	065	050	050	02A	02A	04D
035	066	066	051	051	02B	02B	04E
036	067	067	052	052	02C	02C	04F
037	068	068	053	053	02D	02D	050
038	069	069	054	054	02E	02E	051

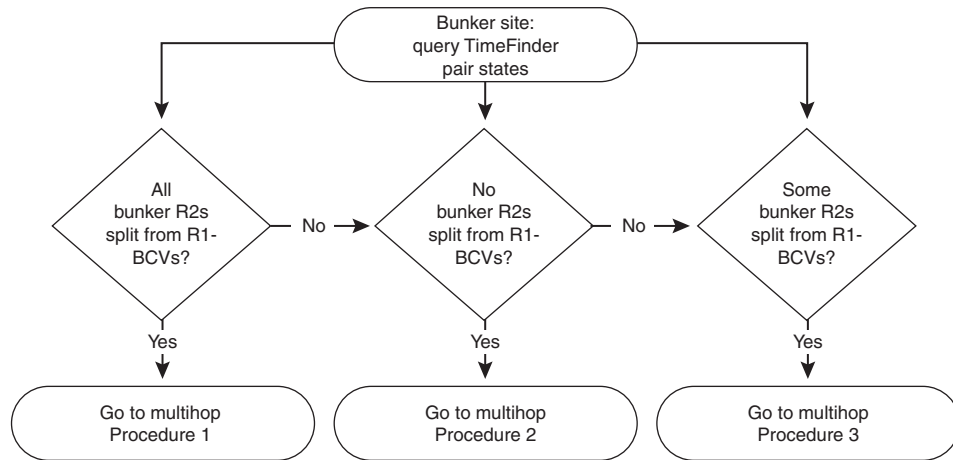
Figure 6-11 Devices Files Used for Multihop Restart Procedures

The following is an example command used to perform the verification of the state of the bunker R2s and BCVs state:

```
symmir -f <DevFile> query -sid <SymmID>
```

The nature of the Symmetrix Automated Replication cycle allows the determination of the devices housing the most current data based on the results of the query command. The flowchart in Figure 6-12 uses these results to direct us to the appropriate restart procedure. There are three possible states of the bunker R2/BCV devices:

- ◆ **All** of the bunker (R2) devices are split from their partner R1-BCV devices
- ◆ **None** of the bunker (R2) devices are split from their partner R1-BCV devices
- ◆ **Some** of the bunker (R2) devices are split from their partner R1-BCV devices

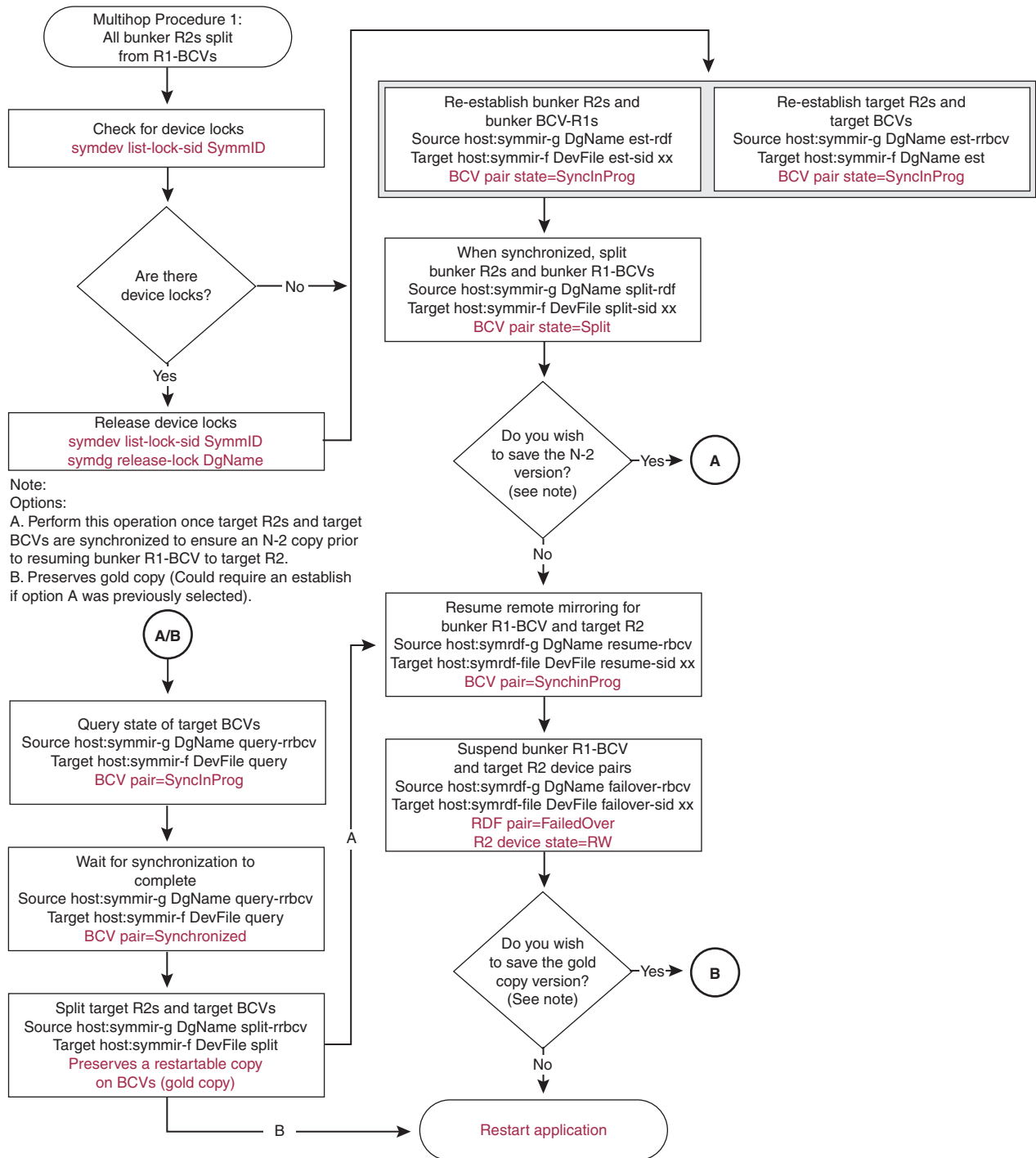


SYM-000614

Figure 6-12 Query Bunker TimeFinder Pair State

6.4.1.1 Procedure 1: All of the Bunker (R2) Devices Are Split from their Partner R1-BCV Devices

The bunker R2s and bunker R1-BCVs being in a split state indicates the dependent write-consistent copy of data is on the R2 devices. This will require the replication process to begin with an establish of the bunker R2s and bunker R1-BCVs. The flowchart in Figure 6-13 details the steps to replicate the most recent copy of data to the target site.



SYM-000615

Figure 6-13 Procedure 1: All Bunker R2s Are Split from their R1-BCVs TimeFinder Pair State

6.4.1.2 Procedure 2: None of the Bunker R2 Devices Being Split from their Partner R1-BCV Devices

If all of the bunker R2s and bunker R1-BCVs are either synchronized or synchronization is in progress, we can deduce the cycle was in the process of copying the most recent copy of data to the bunker R1-BCVs. The flowchart in Figure 6-14 details the steps for this scenario.

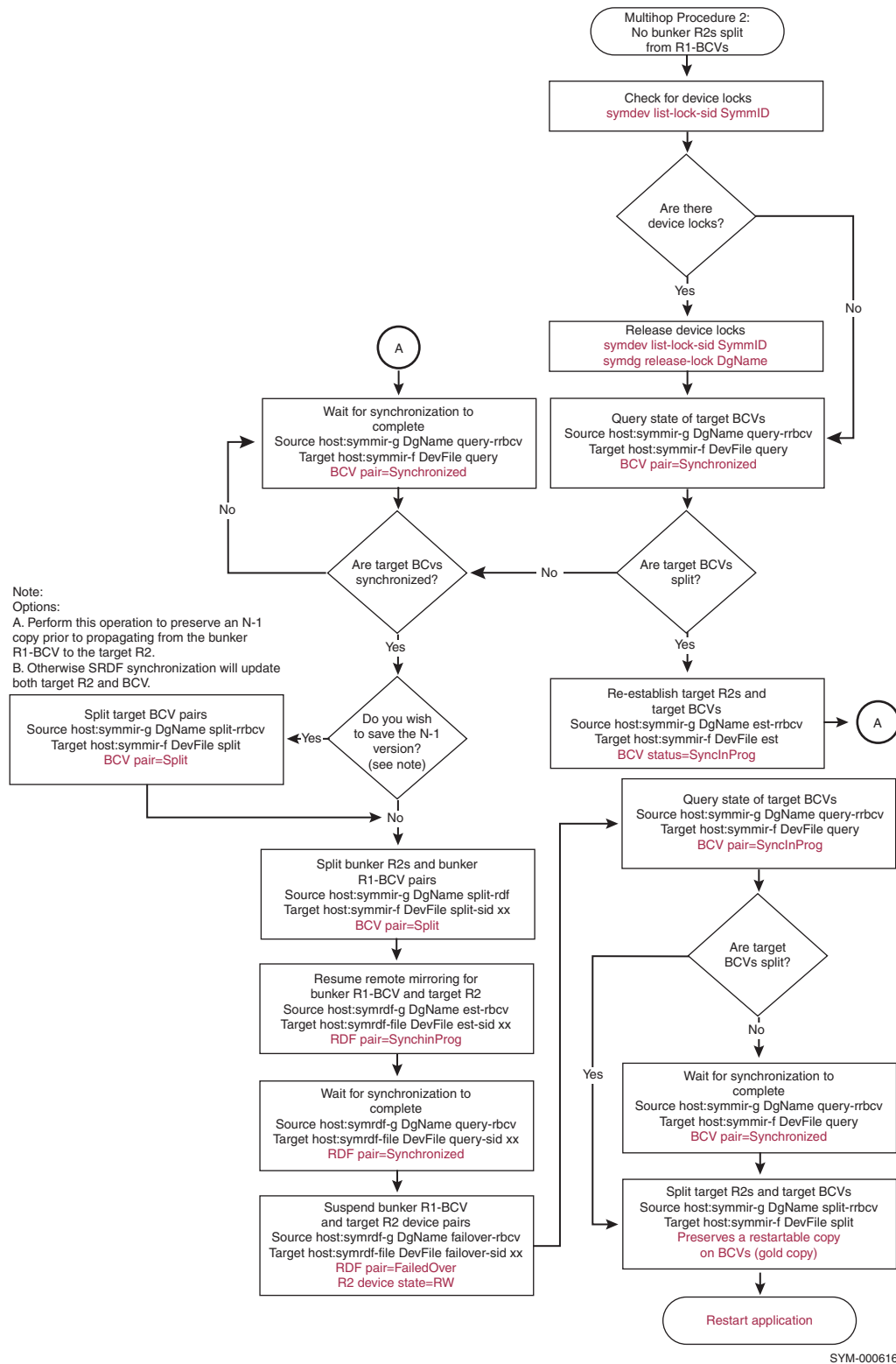


Figure 6-14 Procedure 2: None of the Bunker R2s Being Split from their R1-BCVs TimeFinder Pair State

6.4.1.3 Procedure 3: Some of the Bunker R2 Devices Are Split from their Partner R1-BCV Devices

If the bunker R2s and bunker R1-BCVs are in a mixed state (some synchronized and some split), the cycle was in the process of either establishing or splitting the bunker R1-BCVs from their bunker R2s. The flowchart in Figure 6-15 details the steps for this scenario.

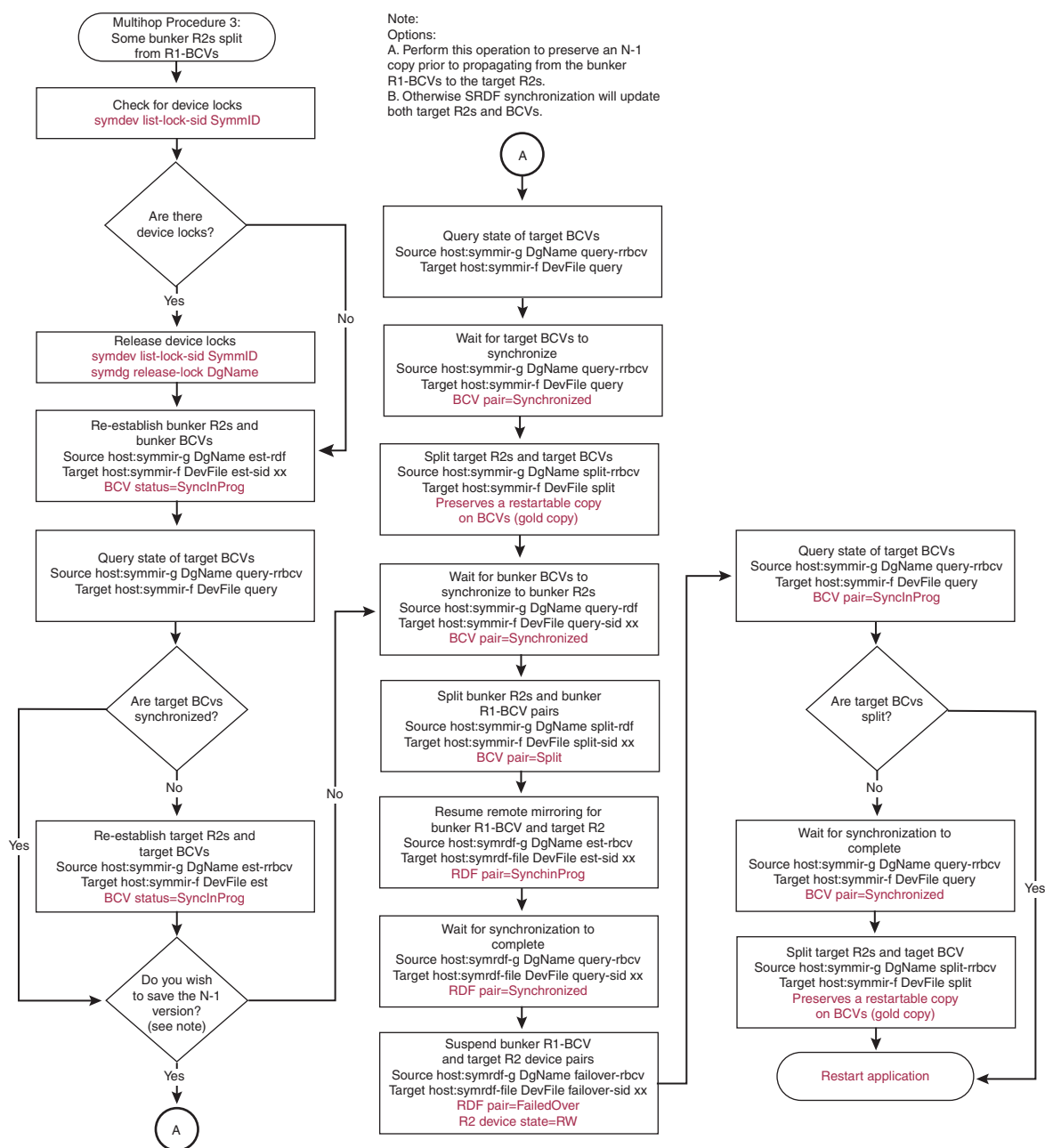


Figure 6-15 Procedure 3: Some of the Bunker R2s Are Split from their R1-BCVs TimeFinder Pair State

6.4.2 Make the Devices Ready to the Host

The R2 devices now contain the most recent restartable copy of data and can be made read/write enabled to the target host. The following command accomplishes this task.

```
symrdf -f <DevFile> -sid <SymmID> -rdfg <RdgGrp> rw_enable r2
```

This command has a similar effect as a `symrdf failover` command, but does not require the SRDF links to be available. The `symrdf failover` command performs a write-disable operation on the R1 devices, suspends the logical link between the R1 and R2 device, and performs the write-enable operation for the R2 devices.

```
symrdf -f <DevFile> -sid <SymmID> -rdfg <RdgGrp> failover
```

The `symdev` command can be used to change the state of individual devices.

```
symdev <SymDevName> -sid <SymmID> rw_enable
```

6.4.3 Restarting the Applications

The environment is now ready to be restarted, which could include the following operations:

- ◆ Import the volumes groups.
- ◆ Perform `fsck` on the file systems.
- ◆ Mount the file systems.
- ◆ Bring up the databases and applications.

Please consult your operating system procedures.

6.4.4 Disaster Restart Off the BCVs

The difference in the procedure(s) to restart off the target R2 devices and the procedure(s) to restart off the target BCV devices is where the most current restartable copy is staged. Once the replication from the bunker site to the target site is complete, the restart will begin from the target BCV devices versus the R2 devices.

There is no reason to read/write enable the R2 devices. The target BCV devices should be protected and addressable to the target host.

This chapter contains the following information:

◆ 7.1 Overview.....	7-2
◆ 7.2 Return Home Overview	7-3
◆ 7.3 SRDF R1 Update/Resynchronization.....	7-4
◆ 7.4 SRDF R1-R2 Swap	7-10
◆ 7.5 Dynamic RDF	7-16

7.1 Overview

Once the issues that necessitated a disaster declaration have been rectified on the primary site, it will be necessary to perform return home procedures. This will require an update to all of the devices in the replication chain in order to return control to the production or primary site and restart the original Symmetrix Automated Replication process.

Considerations for returning home include the length of the outage and the number of invalid tracks accumulated. For example, is the disaster a long-term event or a short defined amount of time? If the event is long in duration, it could require protection be set up to a tertiary site or possibly the original site to provide protection for another unplanned disaster. In shorter outages, simple failback procedures could be used.

7.2 Return Home Overview

This section describes three methods to return home after a disaster in the primary site for single-hop and multihop configurations:

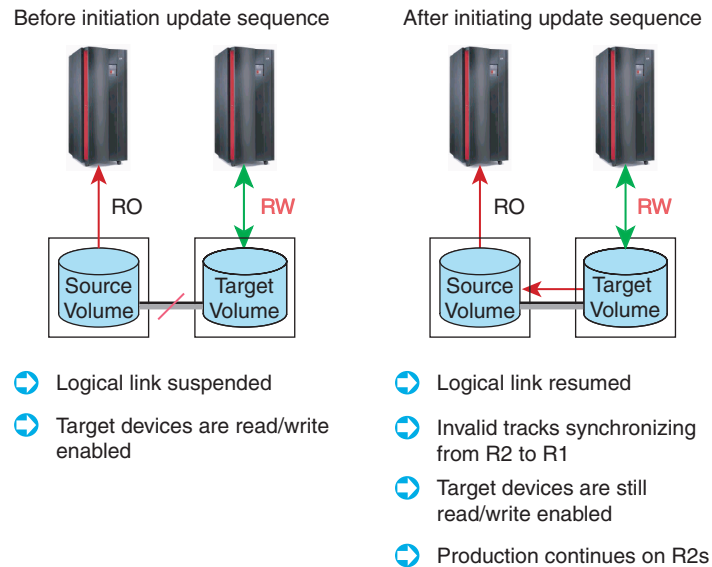
- ◆ Using SRDF R1 update resynchronization
- ◆ Using R1/R2 personality swap (non-dynamic)
- ◆ Using dynamic SRDF reconfiguration

The return home procedures will take into consideration if the production application is running off of the target R2s or target BCVs.

Starting off of	Configuration type	Method		
		R1 Update/resync (sends updates to the R1 while the R2 is being used for production)	R1/R2 swap (swaps the personality of the RDF devices; reverses SRDF/AR direction)	Dynamic RDF (Reverse SRDF/AR direction; provides remote protection during restore)
Target R2s	Single-hop	<i>7.3.2 R1 Update Single-Hop Configuration on page 7-5</i>	<i>7.4.3 R1-R2 Single-Hop Configuration on page 7-11</i>	<i>7.5.5 Dynamic RDF Single-Hop Configuration on page 7-18</i>
	Multihop	<i>7.3.3 R1 Update Multihop Configuration on page 7-7</i>	<i>7.4.4 R1-R2 Multihop Configuration on page 7-13</i>	<i>7.5.6 Dynamic RDF Multihop Configuration on page 7-24</i>
Target BCVs	Single-hop and Multihop	Chapter 7		

7.3 SRDF R1 Update/Resynchronization

This method allows the R1 mirrors, residing at the primary site in a single-hop method, or the R1-BCV mirrors, residing at the bunker site in a multihop method, to be updated with the invalid tracks from the active target site R2s. The production applications continue to run on the R2s on the target site during the R1 invalid track update process. This can significantly reduce the downtime to the applications while doing the final R1/R2 resynchronization. Figure 7-1 demonstrates the host device states and what occurs during an R1 update process.



SYM-000582

Figure 7-1 Host Device States during R1 Update Process

The `symrdf` command is used with a `-until` parameter. The target R2 devices continue to receive updates from production and the R1-BCV devices receive the invalid track updates. The `-until` parameter will continue to monitor the invalid track count owed to the R1-BCVs. Once the track count is less than the number of tracks specified with the `-until` option, the command will exit.

```
symrdf -g <DgName> update -until <NNN>
```

7.3.1 Point of Command Execution

The return home procedures can be issued from either a locally attached host or a host attached to the target site. Depending on the location of the host that is executing the commands (`symmir` and `symrdf`), different SYMCLI device groups (RDF1 and RDF2) and device files are required along with associated command options. The following is an example of commands and required flags issued from the local host. Please consult the *Solutions Enabler Symmetrix SRDF CLI Product Guide* for a complete command structure.

7.3.1.1 TimeFinder Options

Local pairing relationship does not require any options

- ◆ Pairing between Hop 0 and Hop 1
- ◆ Single hop requires `bcv` and `rdf` flag

- symmir -rdf -bcv
- Multihop requires rdf flag
 - symmir -rdf -bcv
- ◆ Pairing between Hop 1 and Hop 2
 - Multihop requires use of a rdf remote bcv flag or a device file designating Symmetrix ID for Hop 2
 - symmir -g <DgName> -rrbcv
 - symmir -f <DevFile> -sid <xxx>

7.3.1.2 SRDF Options

Pairing between Hop 0 and Hop 1:

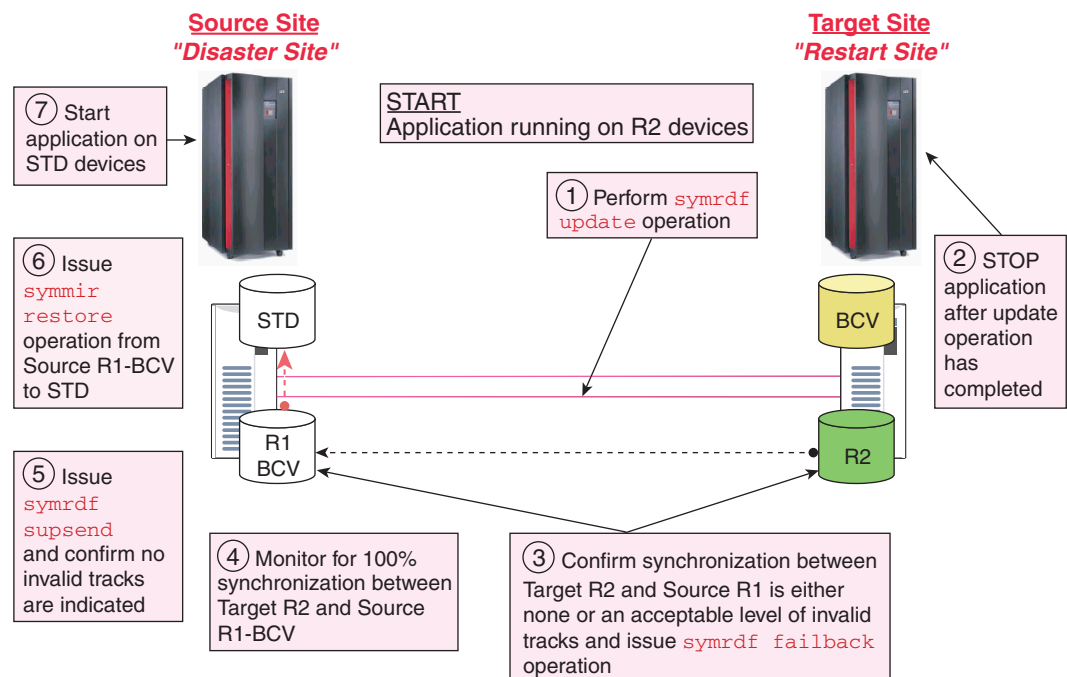
- ◆ Single-hop requires a bcv flag
 - symrdf -bcv
- ◆ Multihop does not require any options

Pairing between Hop 1 and Hop 2:

- ◆ Multihop requires remote bcv flag
 - symrdf -rbcv

7.3.2 R1 Update Single-Hop Configuration

Figure 7-2 is an overview of the steps required to return home in a single-hop configuration.



SYM-000558

Figure 7-2 R1 Update Process Flow for Single-Hop Configurations

The return home commands can be issued from the source site or the target site. The command structure will vary. The steps documented here will provide examples of both. First will be an

example command issued from the source site followed by an example command issued from the target site. The device files used in the commands issued from the target site are the same as in the restart procedures in Chapter 6.

This process begins with the applications running on the target site R2 devices. Prior to issuing the R1 update procedure verify the configuration is in the following state:

Source site STD and R1-BCVs are in a split state.

Source site R1-BCVs are split from the target site R2 devices.

Target site R2s and BCV are in a split state.

The following list details the steps displayed in Figure 7-2:

1. While the production applications are running, begin to propagate the data from the target R2 to the source R1-BCV. This process will happen in the background. This allows the application to be available during the majority of the data movement.

```
symrdf -g <DgName> update -bcv -until <nnn>
symrdf -f <DevFile> update -until <nnn>
```



SYM-000565

2. Once the majority of the data has been propagated to the source site, the applications will need to be brought down on the target site.
3. A failback command can be issued to complete the synchronization from the R2 devices on the target site to the R1-BCVs on the source site. The failback command will also put the devices in the desired state, such as R2 devices in a read only state.

```
symrdf -g <DgName> failback -bcv
symrdg -f <DevFile> failback
```

4. Wait for complete synchronization and failback.

```
symrdf -g <DgName> query -bcv
symrdf -f <DevFile> query
```

5. Suspend the link between the R1-BCVs and the R2s.

```
symrdf -g <DgName> suspend -bcv
symrdf -f <DevFile> suspend
```



SYM-000565

6. Issue a command to perform a restore from the R1-BCVs to the source STD volumes.

```
symmir -g <DgName> restore
symmir -f <DevFile> restore -rdfg <xx> -sid <xxx>
```

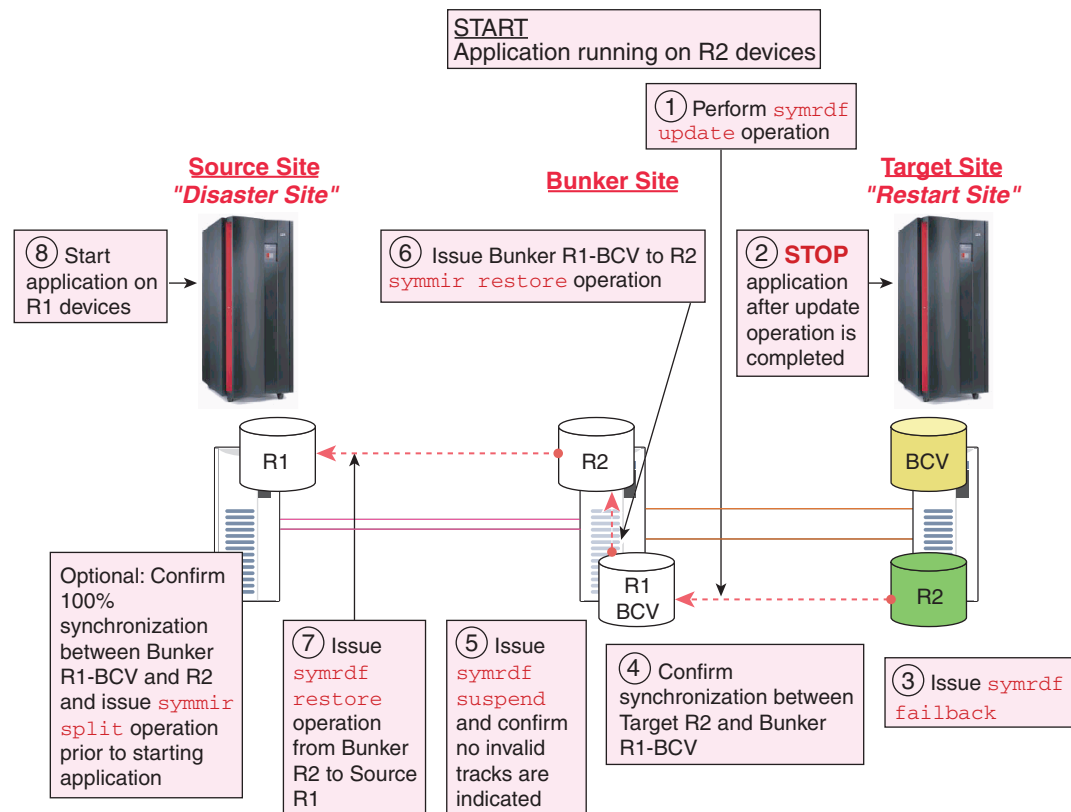


SYM-000566

7. The operating systems, databases and applications can be brought up on the source STD devices immediately after restore command issued. The Symmetrix will retrieve the required information from the R1-BCV if needed.

7.3.3 R1 Update Multihop Configuration

Figure 7-3 is an overview of the steps required to return home in a multihop configuration.



SYM-000559

Figure 7-3 R1 Update Process Flow for Multihop Configurations

The return home commands can be issued from the source site or the target site. The command structure will vary. The steps documented here will provide examples of both. First will be an example command issued from the source site followed by an example command issued from the target site. The device files used in the commands issued from the target site are the same as in the restart procedures in Chapter 6.

This process begins with the applications running on the target site R2 devices. Prior to issuing the R1 update procedure verify the configuration is in the following state:

- ◆ Bunker site R2s and R1-BCVs are in a split state.
- ◆ Bunker site R1-BCVs are split from the target site R2 devices.
- ◆ Target site R2s and BCV are in a split state.

The following list details the steps displayed in Figure 7-3:

1. While the production applications are running, begin to propagate the data from the target R2 to the bunker R1-BCV. This process will happen in the background. This allows the application to be available during the majority of the data movement.

```
symrdf -g <DgName> update -rbcv -until <nnn>
symrdf -f <DevFile> update -until <nnn>
```



SYM-000567

2. Once the majority of the data has been propagated to the bunker site, the applications will need to be brought down on the target site.
3. A failback command can be issued to complete the synchronization from the R2 devices on the target site to the R1-BCVs on the bunker site. The failback command will also put the devices in the desired state, such as R2 devices in a read-only state.

```
symrdf -g <DgName> failback -rbcv
symrdf -f <DevFile> failback
```

4. Wait for complete synchronization and failback.

```
symrdf -g <DgName> query -rbcv
symrdf -g <DevFile> query
```

5. Suspend the link between the R1-BCVs at the bunker site and the R2s at the source site.

```
symrdf -g <DgName> suspend -rbcv
symrdf -f <DevFile> suspend -rdfg <xx> -sid <xxx>
```



SYM-000568

6. Issue a command to perform a restore from the bunker R1-BCVs to the bunker R2 devices.

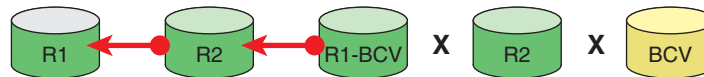
```
symmir -g <DgName> restore -rdf
symmir -f <DevFile> restore -rdfg <xx> -sid <xxx>
```



SYM-000569

7. Issue the restore of bunker R2 devices to the source site R1 devices. This can occur in conjunction with step 6.

```
symrdf -g <DgName> restore
symrdf -f <DevFile> restore -rdfg <xx> -sid <xxx>
```



SYM-000570

Optional: Wait for 100 percent synchronization between bunker R1-BCVs and bunker R2 devices. Split the relationship between the bunker R1-BCVs and the bunker R2 devices. This will create a restore point at the bunker site as well as the target site.

```
symmir -g <DgName> split -rdf
symmir -f <DevFile> split -rdfg <xx> -sid <xxx>
```



8. The operating systems, databases, and applications can be brought up immediately after restore command is issued. The Symmetrix will retrieve the required information from the R2 if needed.

7.4 SRDF R1-R2 Swap

Returning home using the SRDF R1-R2 swap is an alternative to the update/R1 Resynchronization option. The major difference between the R1 update option and the R1-R2 swap option is that after using the `symrdf swap` command, the resynchronization is a continuous operation. The R1 update option could require multiple resynchronization cycles. The R1-R2 swap method changes the RDF personality of the RDF devices of the specified device group. Therefore, the R1-R2 swap option is recommended when the environment has a high number of invalid tracks and also generates a high number of invalid tracks per interval of time. This procedure starts with the production applications running off of the target R2s.

7.4.1 R1-R2 Swap Requirements

To use R1-R2 swap, one of the following three scenarios must apply to the configuration:

- ◆ SRDF and Configuration Manager licenses with Enginuity Version 5x66 or 5267
- ◆ SRDF and Configuration Manager licenses with Enginuity Version 5567 or higher and the devices to be swapped are not Dynamic RDF enabled in your Symmetrix configuration
- ◆ SRDF license with Enginuity Version 5567 or higher and Dynamic RDF is enabled in your Symmetrix configuration
 - Enginuity 68, the Dynamic RDF configuration state of the Symmetrix must be enabled.
 - RDF1 capable
 - RDF2 capable
 - RDF1 and RDF2 capable
 - R1/R2 swap operations are not supported for Enginuity Version 5669 configurations where the R2 device is larger than the R1 device.

An SRDF extended-distance configuration that uses RFA (remote fibre adapter) supports the R1-R2 swap command.

An SRDF extended-distance configuration that uses RA (ESCON adapter) supports only the director R1-R2 swap where the personalities of the adapter must be explicitly changed.

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

7.4.2 R1-R2 Swap Command Format

When the applications are running off the R2s (that is, when the R2 devices are ready and read/write enabled), the SRDF operation cannot be resumed for data protection reasons. The swap option of the `symrdf` command swaps the SRDF relationship between the two SRDF devices (that is, the R1s become R2s and the original R2s become R1s). After the swap, the SRDF operation can be resumed without interrupting the applications. The following is the command format:

```
symrdf -g <DgName> [-bcv|-all] [-bypass] swap [-refresh R1|R2]
```

`[-bcv]` – Target RDF Hop 1 BCV associated devices for the swap

`[-all]` – Target both standard RDF device and locally attached BCV RDF devices

`[-bypass]` – Use to bypass Symmetrix locks (CAUTION)

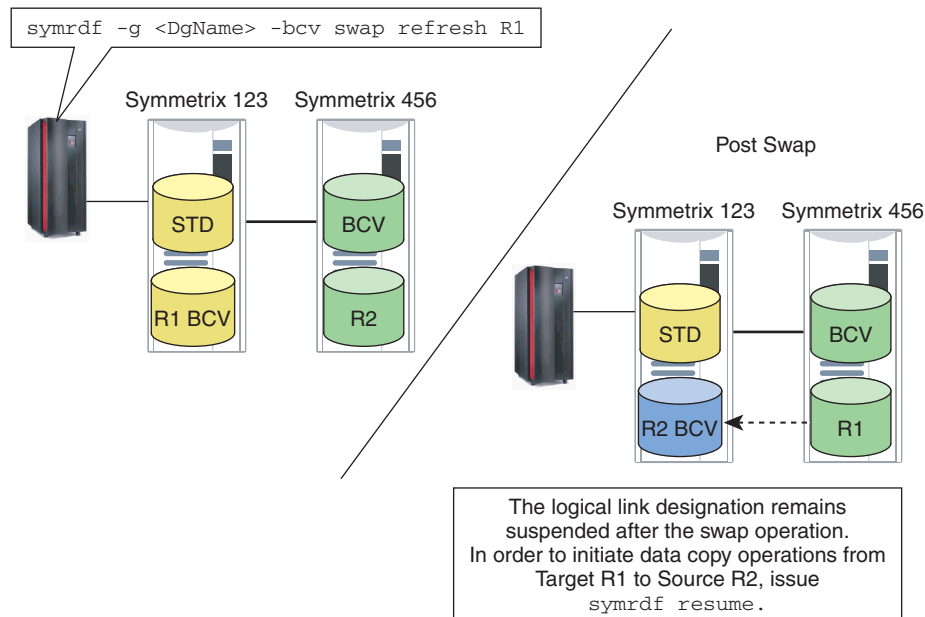
`[-refresh]` – Marks R1 or R2 device(s) to refresh from its remote mirror

Determine which SRDF device holds the valid copy of data before the refresh action. If you specify `refresh R1|R2`, it will indicate device(s) do **not** hold a valid copy *before the swap operation begins*.

The `refresh R2` option indicates the R1 device(s) hold the valid copy and the R2 device's invalid tracks will be updated using the (original) R1 data.

The `refresh R1` option indicates the R2 device(s) hold the valid copy and the R1 device's invalid tracks will be updated using the (original) R2 data.

Figure 7-4 is an example of the of the `symrdf swap -refresh R1` command.

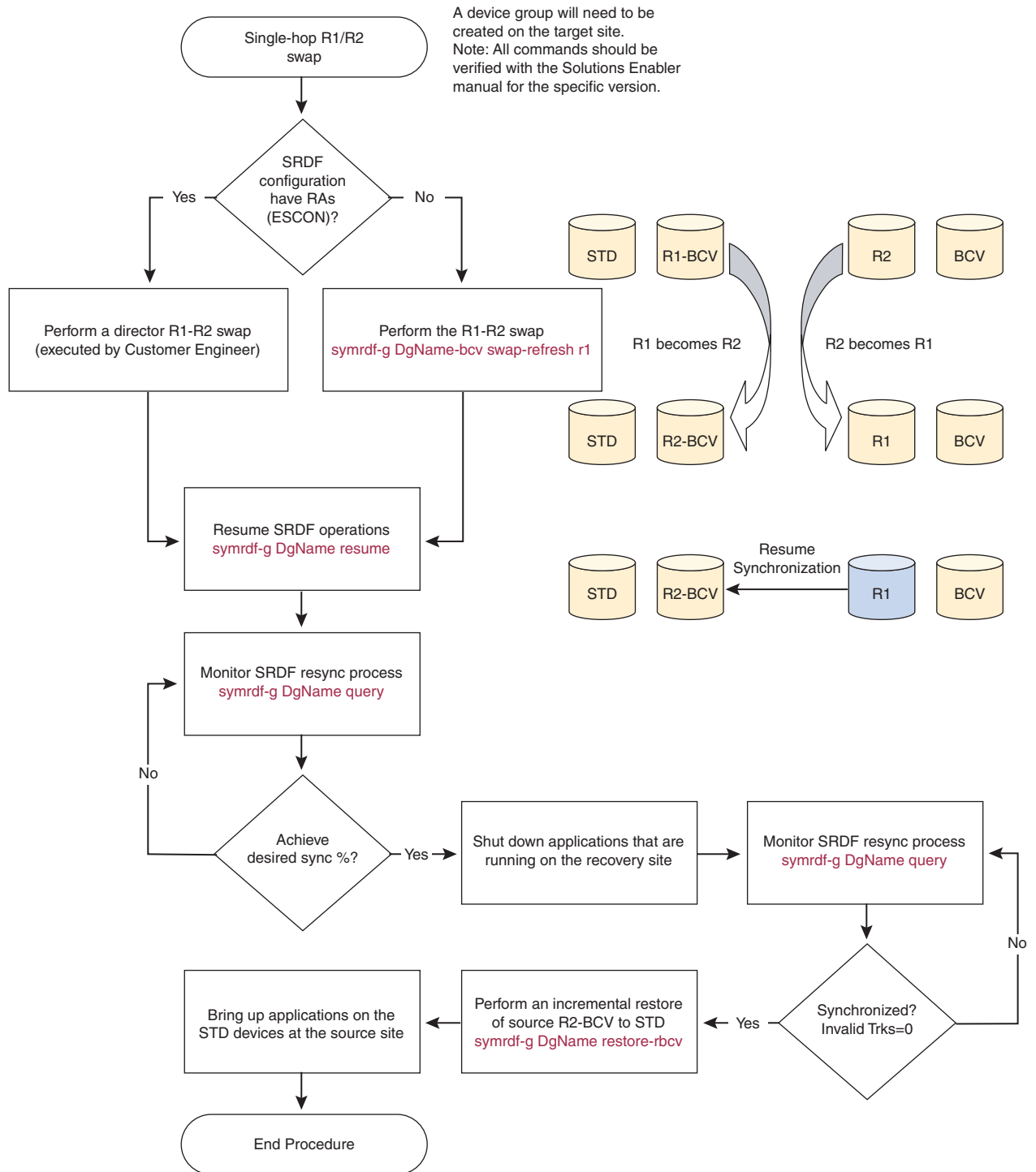


SYM-000564

Figure 7-4 Example of `symrdf swap` with Resume Option

7.4.3 R1-R2 Single-Hop Configuration

Figure 7-5 demonstrates the steps for performing the R1-R2 swap for a single-hop configuration. These procedures are to be issued from the recovery site. A device group will need to be created.



SYM-000618

Figure 7-5 Single-Hop R1-R2 Swap Procedures

Once these procedures are complete another R1-R2 swap will need to be performed to reconfigure the devices to the correct attributes. Figure 7-6 illustrates this process.

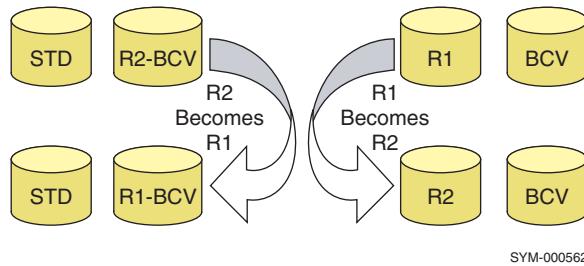
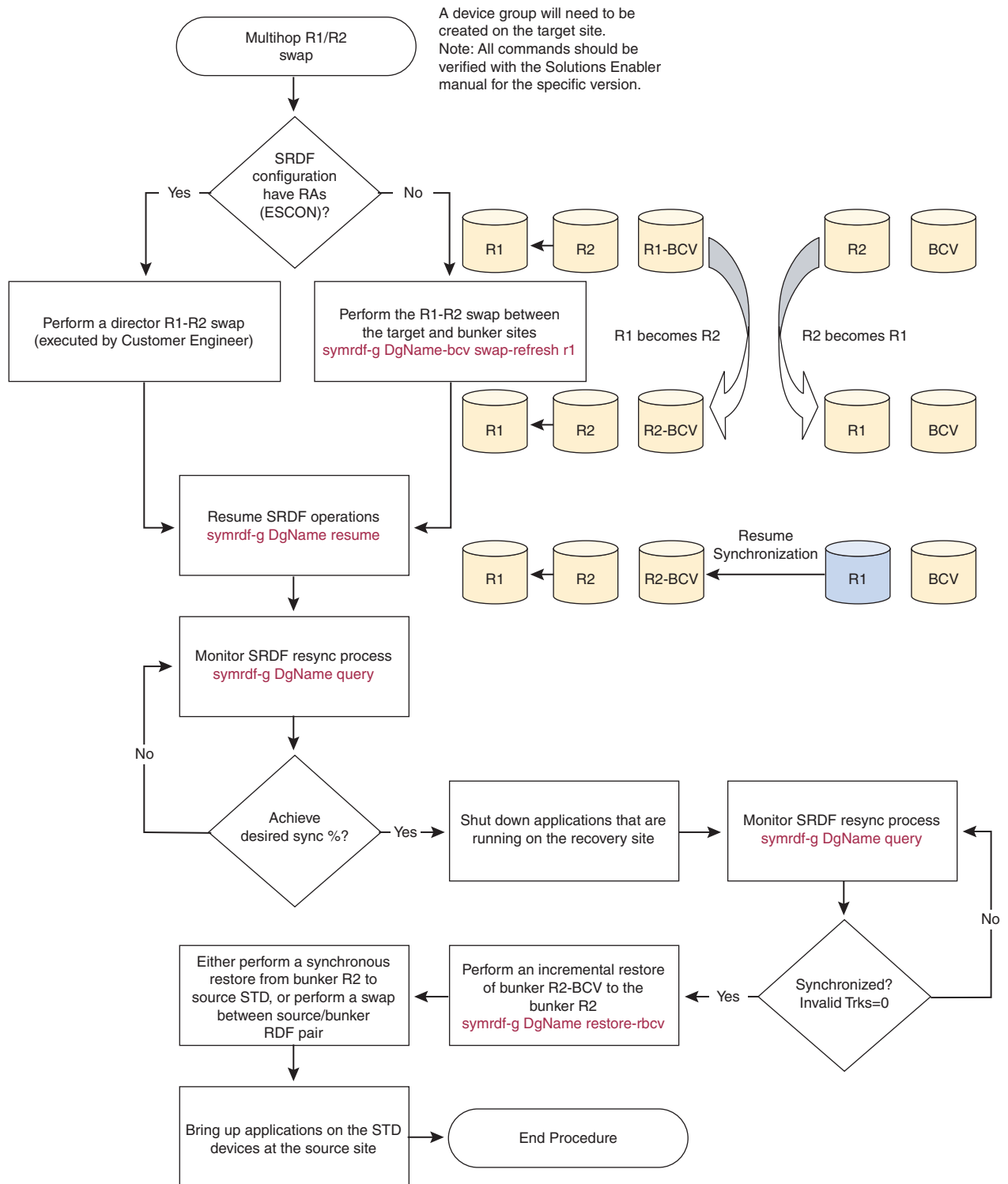


Figure 7-6 R1-R2 Reconfiguration of Devices for Single Hop

The SRDF/AR procedure can now be started, without performing full establishes. The configuration will need to be in the correct state in order to start the process.

7.4.4 R1-R2 Multihop Configuration

Figure 7-7 demonstrates the steps for performing the R1-R2 swap for a multihop configuration. These procedures are to be issued from the recovery site. A device group will need to be created.



SYM-000619

Figure 7-7 Multihop R1-R2 Swap Procedures

Once these procedures are complete another R1-R2 swap will need to be performed to reconfigure the devices to the correct attributes. Figure 7-8 illustrates this process.

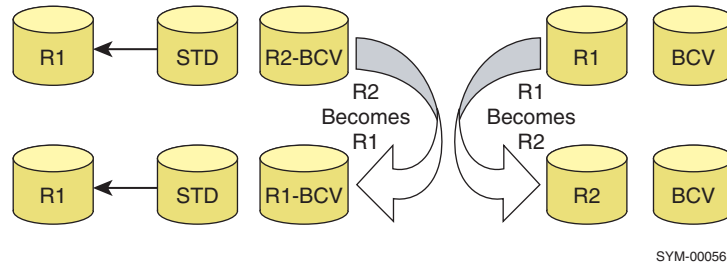


Figure 7-8 Reconfiguration of Devices for Multihop

The SRDF/AR procedure can now be started, without performing full establishes. The configuration will need to be in the correct state in order to start the process.

7.5 Dynamic RDF

Using Dynamic SRDF it is possible to reconfigure the SRDF device relationships. This can create a new SRDF/AR definition to propagate data from the target site to the source site or to a tertiary site. This process provides data protection in case of another disaster.

7.5.1 Dynamic RDF Requirements

The following is a list of licensing and Enginuity requirements:

- ◆ SRDF licenses with Enginuity Version 5568 or higher
 - Management via the command line requires Configuration Manager license
 - Management via ControlCenter requires the Symmetrix Manager license
- ◆ Dynamic RDF configuration state of the Symmetrix must be enabled
 - Can be done from SYMCLI or service processor
- ◆ Devices must be designated as dynamic RDF_capable
 - SYMCLI or ControlCenter
 - dyn_rdf1_only
 - dyn_rdf1_only
 - dyn_rdf
 - Service Processor
 - DR1
 - DR2
 - DRx
- ◆ RDF links must be suspended prior to creating or deleting SRDF pairs

7.5.2 Verifying Dynamic RDF Devices

In order to verify the devices have been configured as dynamic RDF capable, use the `symdev list` command with the `-dynamic` option as follows:

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

7.5.3 Creating a Device File

A device file is necessary to create RDF pairs. 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 file in the first column is an R1 or an R2. These devices in the first column associate with the specified Symmetrix ID.

```
# Device File
09C 054
09D 055
09E 056
```

7.5.4 Dynamic RDF Command Format

The creation of a dynamic SRDF pair utilizes the `createpair` option of the `symrdf` command. Likewise the deletion of a dynamic SRDF pair utilizes the `deletepair` option of the `symrdf` command. The following is the command format:

```
symrdf -f <DevFile> -sid xxx -RDFG xx createpair [-g DgName] -type [RDF1|RDF2]
```

`-establish` Invalidates R2s, mergers track tables, brings up RDF links, starts copy process from R1 to R2

`-restore` Invalidates R1s, mergers track tables, brings up RDF links, starts copy process from R2 to R1

`-invalidate` Invalidates [R1 | R2] in the device file, but does not bring up the RDF links and initiate data copy

Establish will require `-invalidate R2`

Restore will require `-invalidate R1`

```
symrdf -f <DevFile> -sid xxx -RDFG xx deletepair [-g DgName]
```

The `-g DgName` option will create a device group including the devices listed in the device file. Future `symrdf` commands can reference the device group versus the device file.

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

The device is in one of the following BCV pair states: Synchronized, SyncInProgress, Restored, RestoreInProgress, SplitInProgress.

The device is the source or target of an EMC snap operation.

There is a background BCV split operation in progress.

Devices are in the back-end not ready state.

There is an optimizer swap in progress on a device.

The emulation type is not the same.

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 operation involves one or more of the following unsupported devices:

- VCM DB
- SFS
- RAD
- DRV
- RAID_S
- WORM-enabled devices
- Four-way mirror
- Meta member
- PPRC mainframe system

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

The device is in one of the following BCV pair states: Synchronized, SyncInProgress, Restored, RestoreInProgress, SplitInProgress.

The device is the source or target of an EMC snap operation.

There is a background BCV split operation in progress.

Devices are in the back-end not ready state.

There is an optimizer swap in progress on a device.

The emulation type is not the same.

The SRDF pairs are in one of the RDF-in-progress states.

There are existing local invalid tracks on either the local or remote device.

There are remote invalid tracks and the `-force` option was not specified.

RDF Consistency is enabled.

The links are not suspended.

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

Suspended

Split

Failed over

7.5.5 Dynamic RDF Single-Hop Configuration

The single-hop procedures are split into five steps:

1. Reconfigure SRDF to run SRDF/AR from target to (new) source site.
2. Redefine and start new SRDF/AR process.
3. Swap applications from the target site to the (new) source site.
4. Reconfigure SRDF to run SRDF/AR from (new) source site to target site.
5. Redefine and start new SRDF/AR process.

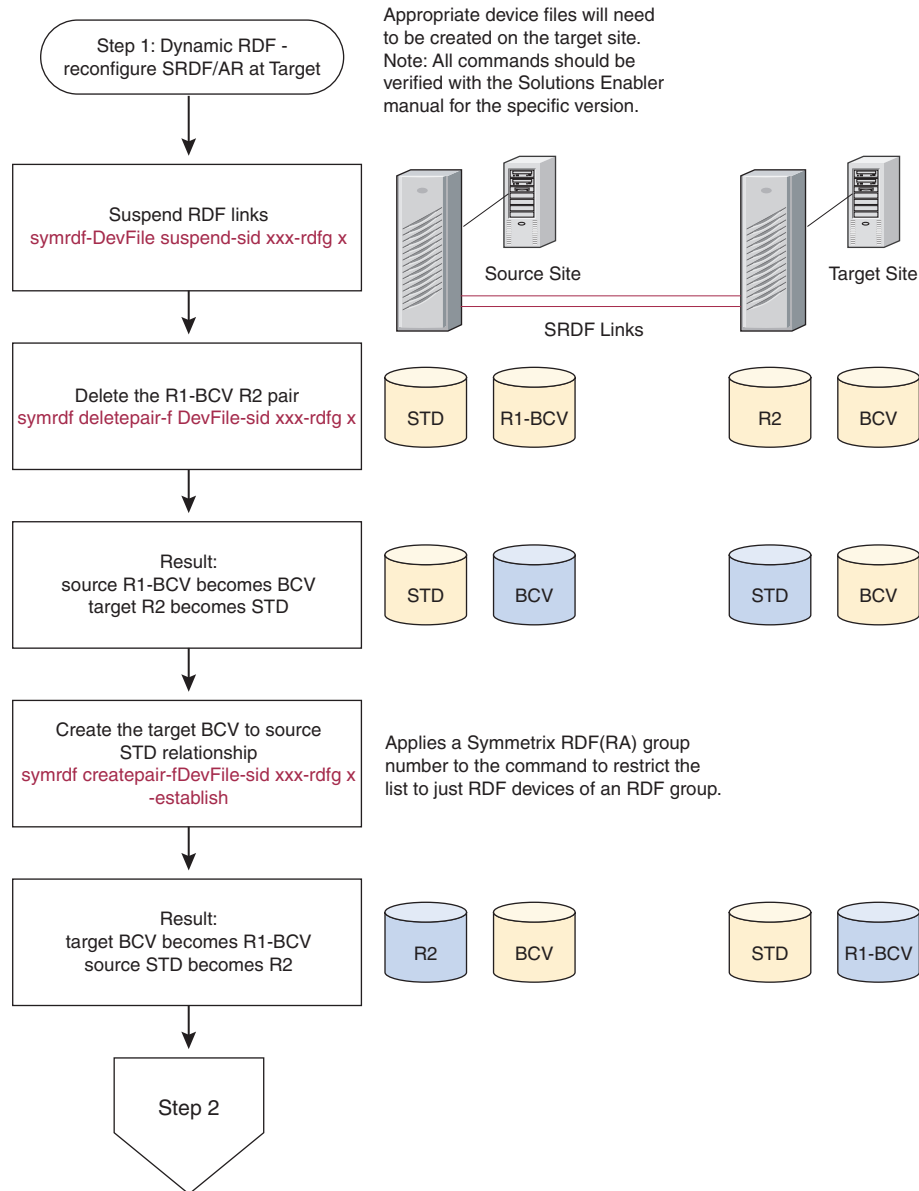
This return home procedure starts with the applications running off the R2 devices.

7.5.5.1 Step 1: Reconfigure SRDF to Run SRDF/AR from Target to (new) Source Site

Using Dynamic SRDF, we can configure/re-configure the SRDF device relationships to either:

1. Create a new SRDF/AR definition to propagate data from the target site to the source site.
2. Establish a relationship to a tertiary site in a different location.

This will provide protection from another disaster while you are failed over to the current recovery (target) site. Figure 7-9 demonstrates this process.



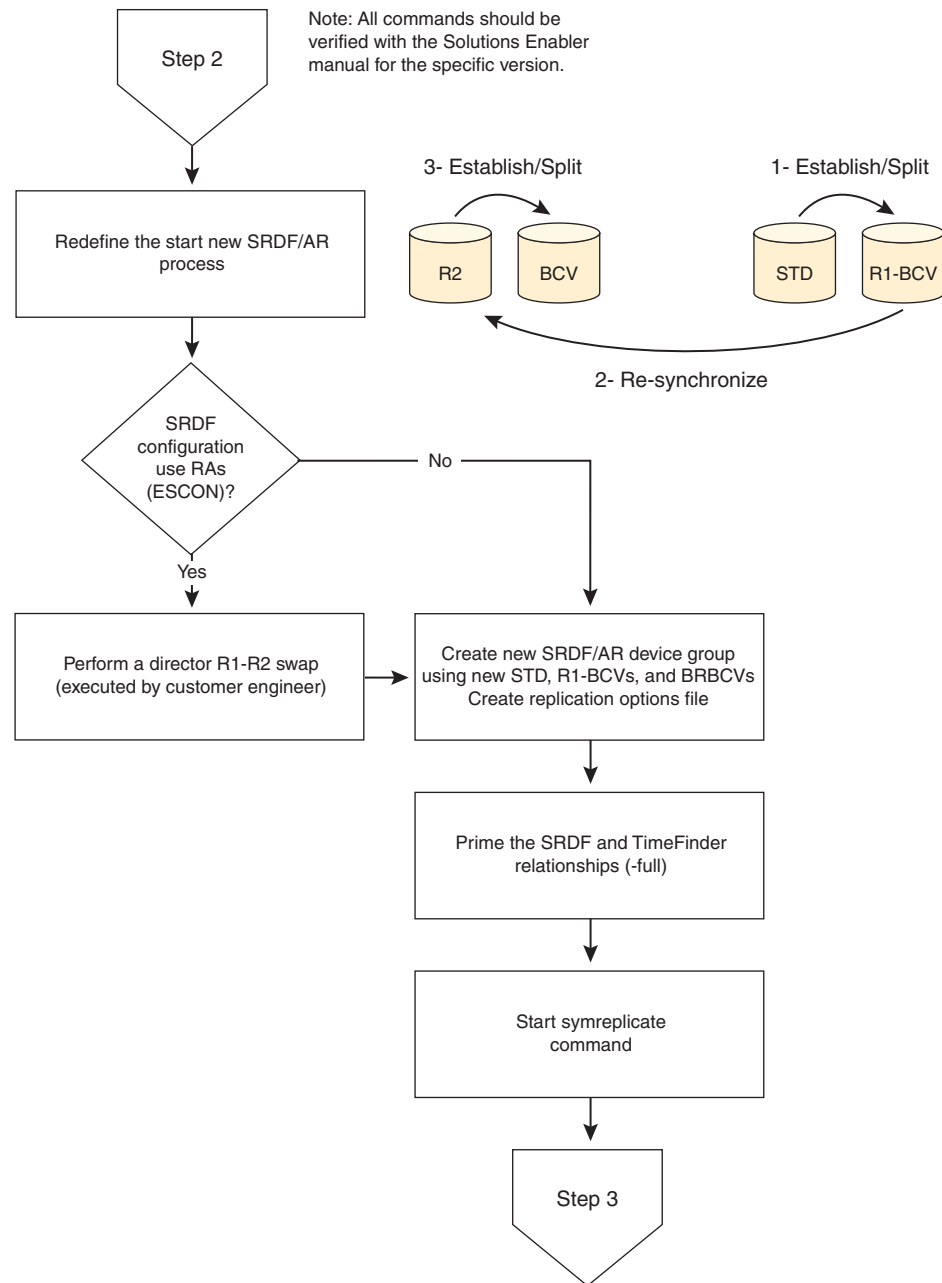
SYM-000620

Figure 7-9 Reconfigure SRDF

7.5.5.2 Redefine and Start New SRDF/AR Process

In Figure 7-10 we define a new SRDF/AR process using the hardware configuration created in step 1. This new process must be initialized as any other new SRDF/AR process.

Note: An SRDF configuration using ESCON adapters requires a director R1-R2 swap to be executed by the Customer Engineer.



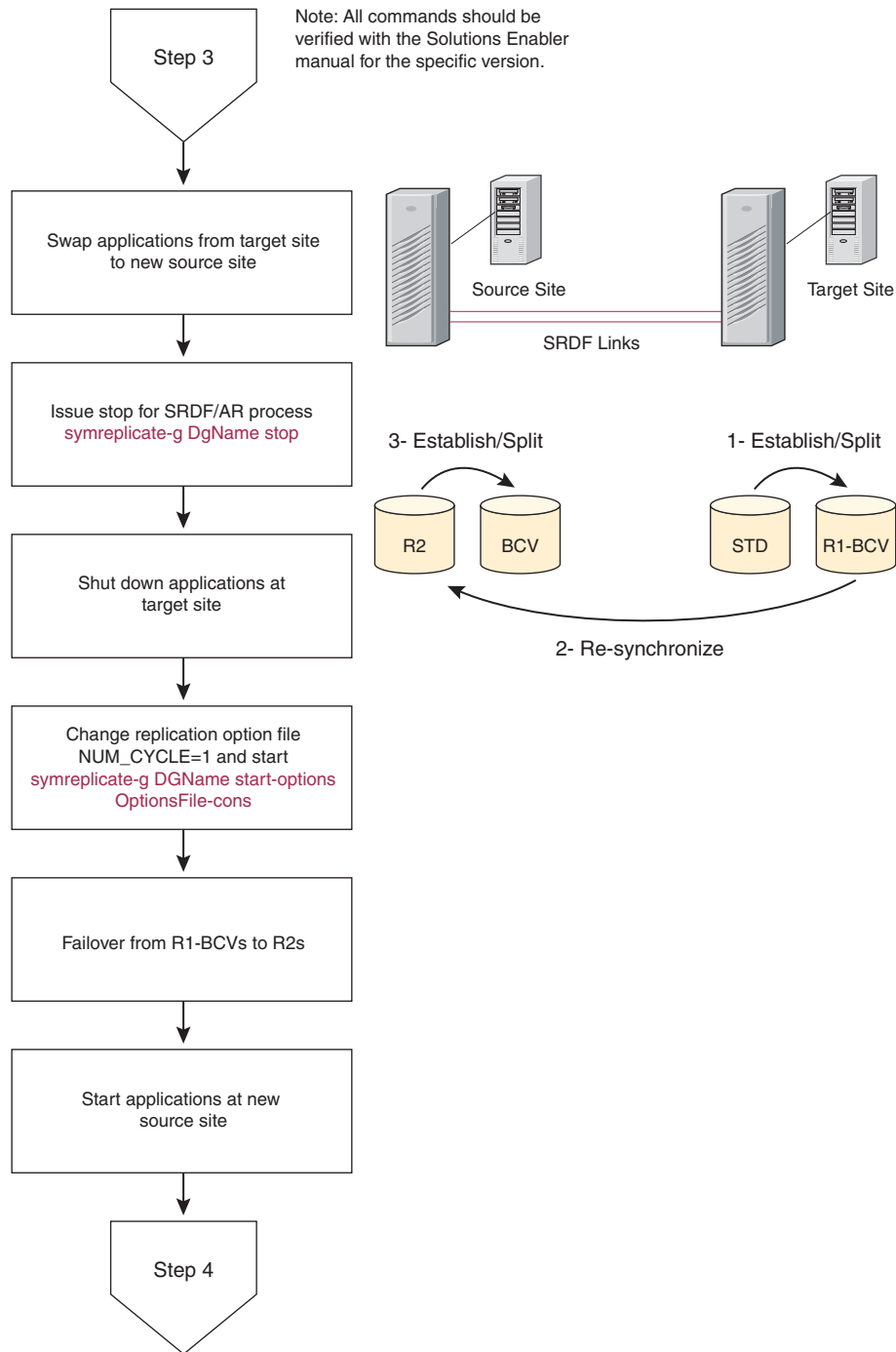
SYM-000621

Figure 7-10 Redefine and Start the new SRDF/AR Process

At the end of step 2, the remote replication runs, sending data from the target site to the new source site used in the configuration defined in step 1.

7.5.5.3 Step 3: Swap the Application from the Target Site to the new Source Site

Step 3 starts with the SRDF/AR process running. The purpose of step 3 is to swap the production application from the target site to the new source site with a minimum amount of downtime. Applications are running off the target STDs and SRDF/AR is performing the data replication from the target to the source site. Figure 7-11 demonstrates this process.



SYM-000622

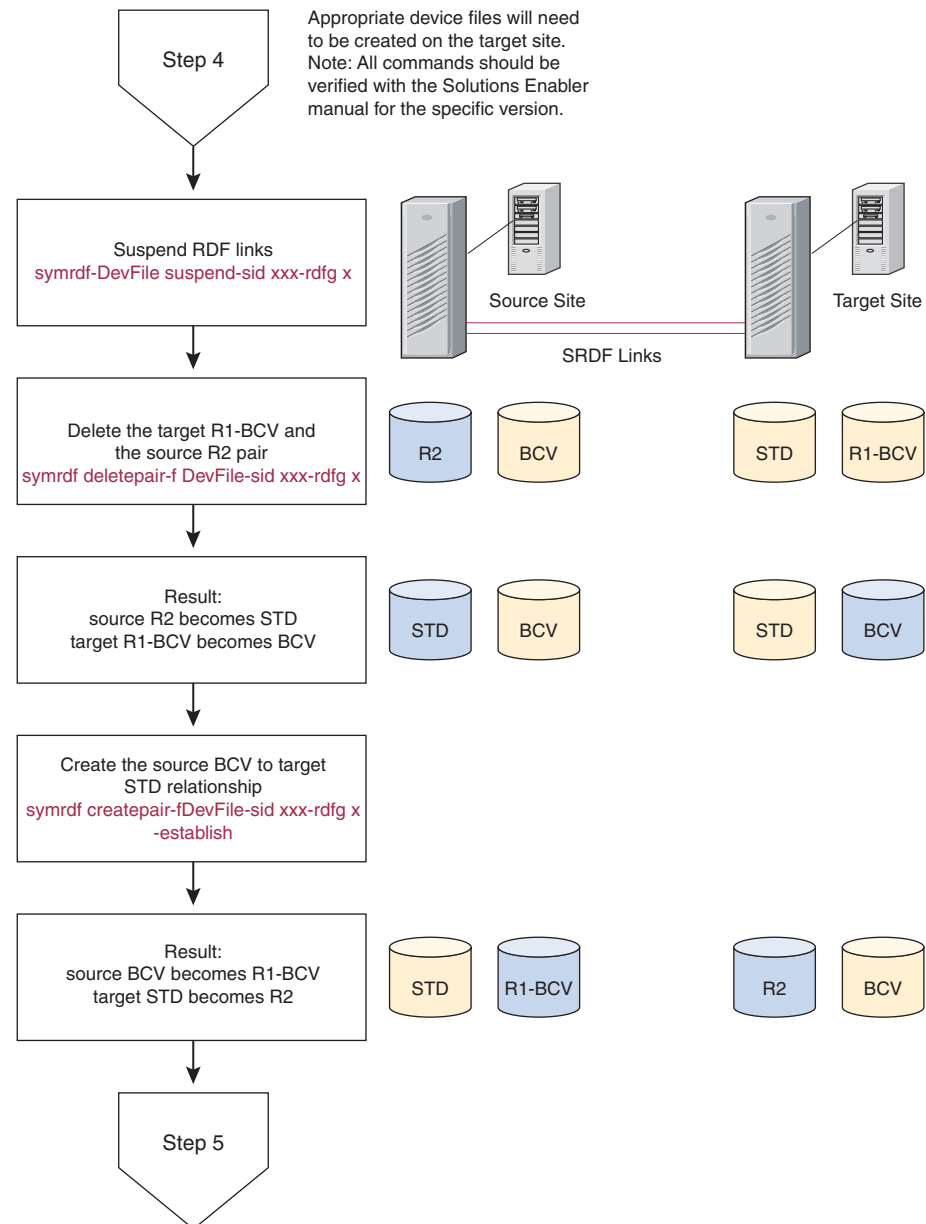
Figure 7-11 Swap the Application

Note that most of the downtime of the applications will be measured by the time that it takes to execute the final cycle of the data replication after the applications are stopped.

At the end of step 3, the production applications are running in the source site again (or in a new site) and the target site is inactive. To reactivate the remote copy solution, steps 4 and 5 must be executed.

7.5.5.4 Step 4: Reconfigure SRDF to Run SRDF/AR from Source to Target Site

Step 4 starts with the production applications running off the R2s located in the source site. The purpose of this step is to delete the existing SRDF configuration and create a new one with STD and R1-BCVs in the source site and R2 and BCVs in the target site. The production applications are not affected by the execution of the SRDF configuration commands. This is outlined in Figure 7-12.



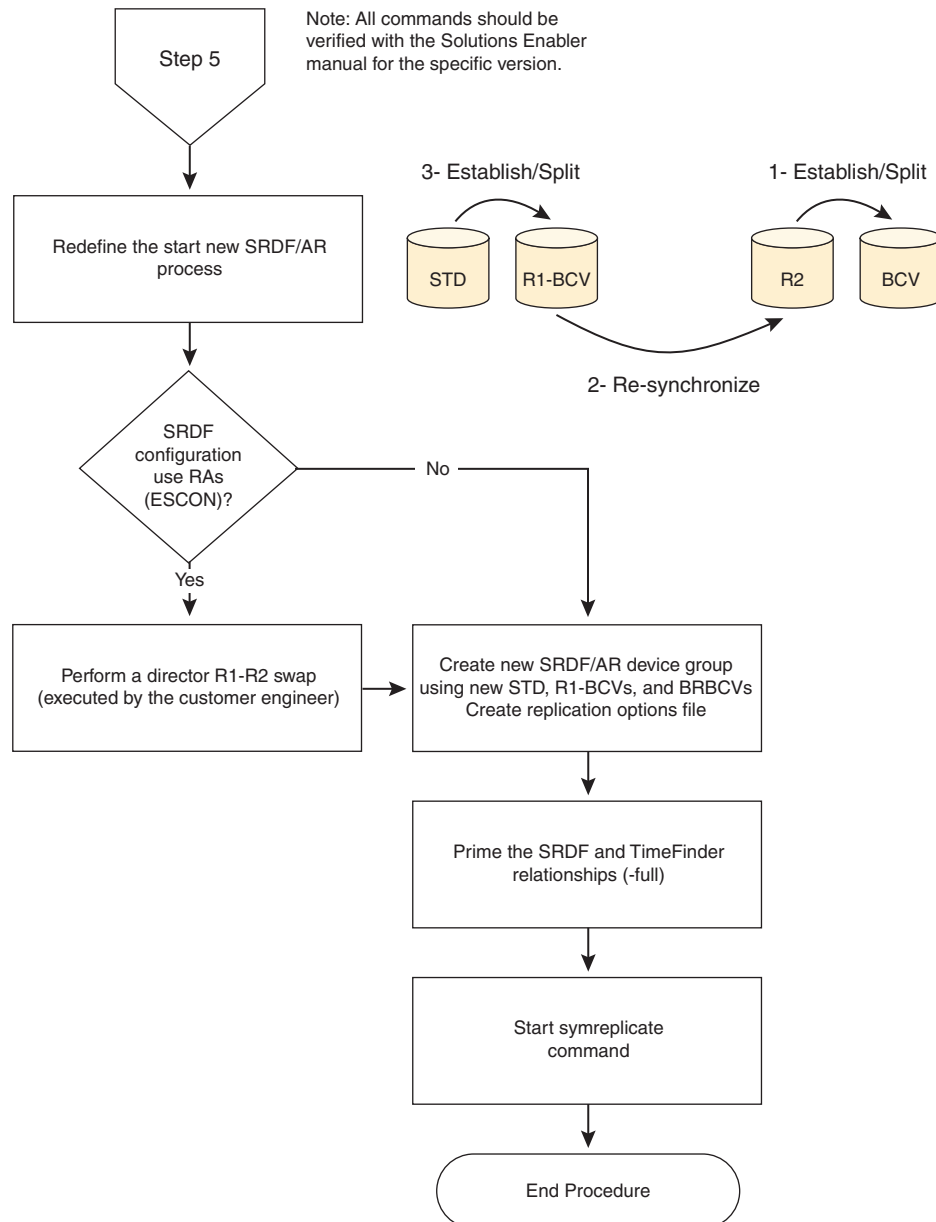
SYM-000623

Figure 7-12 Reconfigure SRDF

7.5.5.5 Redefine and Start SRDF/AR Process

Step 5 is similar to step 2 except for the devices that are part of the SRDF configuration. Because the SRDF relationships are new, a full synchronization will be required. See Figure 7-13.

Note: An SRDF configuration using ESCON adapters requires a director R1-R2 swap to be executed by the Customer Engineer.



SYM-000624

Figure 7-13 Redefine and Start SRDF/AR Process

7.5.6 Dynamic RDF Multihop Configuration

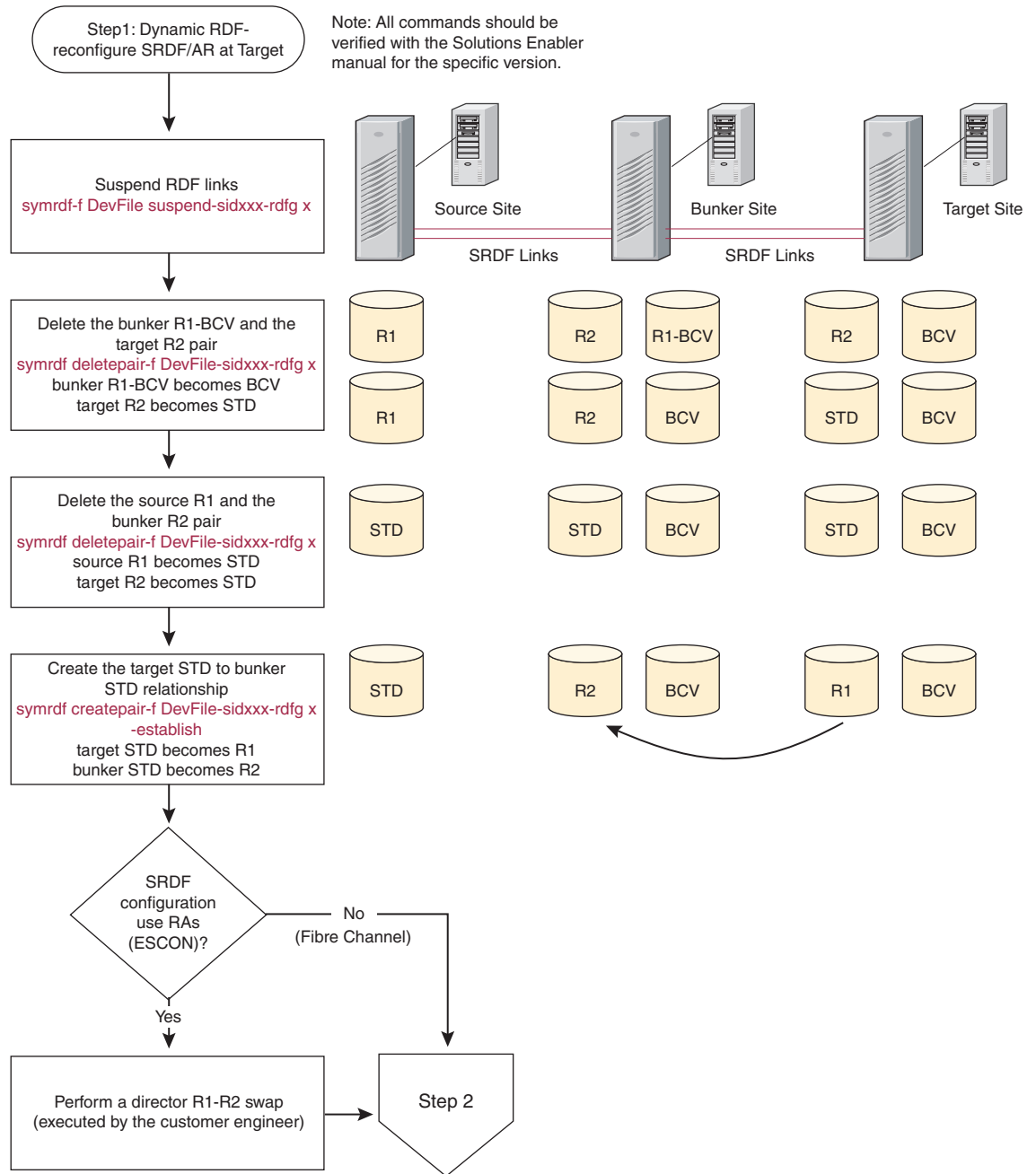
The following return home procedures assume that the same network configuration used before the disaster will be used to return home. The multihop procedures are split into three steps:

1. Reconfigure process flow.
2. Resynchronization first from target site to bunker site. Followed by bunker site to the source site.
3. Restart applications, reconfigure SRDF, and restart SRDF/AR.

This return home procedure starts with the applications running off the R2 devices at the target site.

7.5.6.1 Step 1: Reconfigure Process Flow

Using Dynamic SRDF, we can configure/re-configure the SRDF device relationships. Figure 7-14 demonstrates this process.

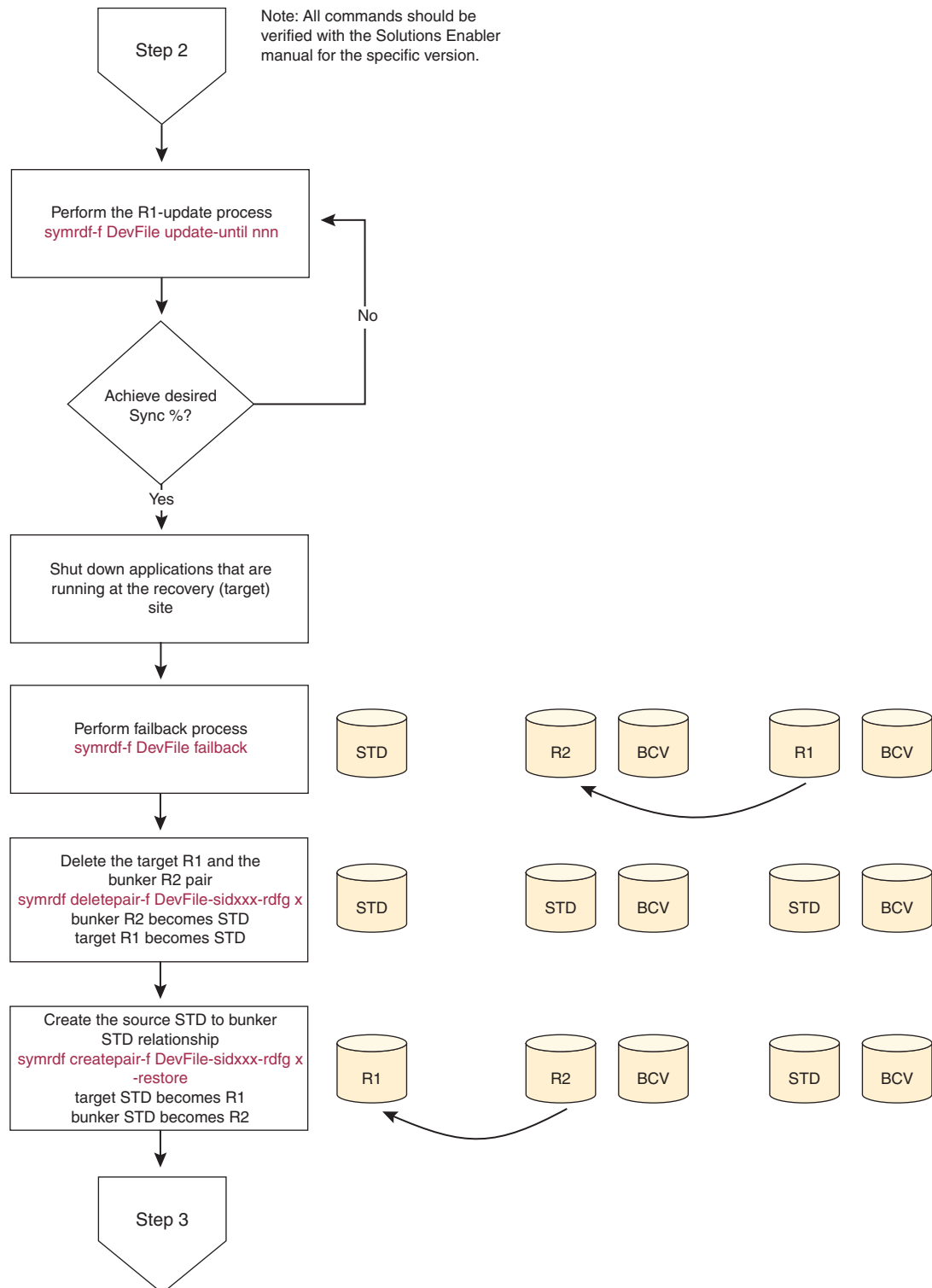


SYM-000625

Figure 7-14 Redefine and Start SRDF/AR Process

7.5.6.2 Step 2: Resynchronization First from Target Site to Bunker Site, Followed by Bunker Site to the Source Site

Figure 7-15 demonstrates how to propagate the data from the target site to the source site.

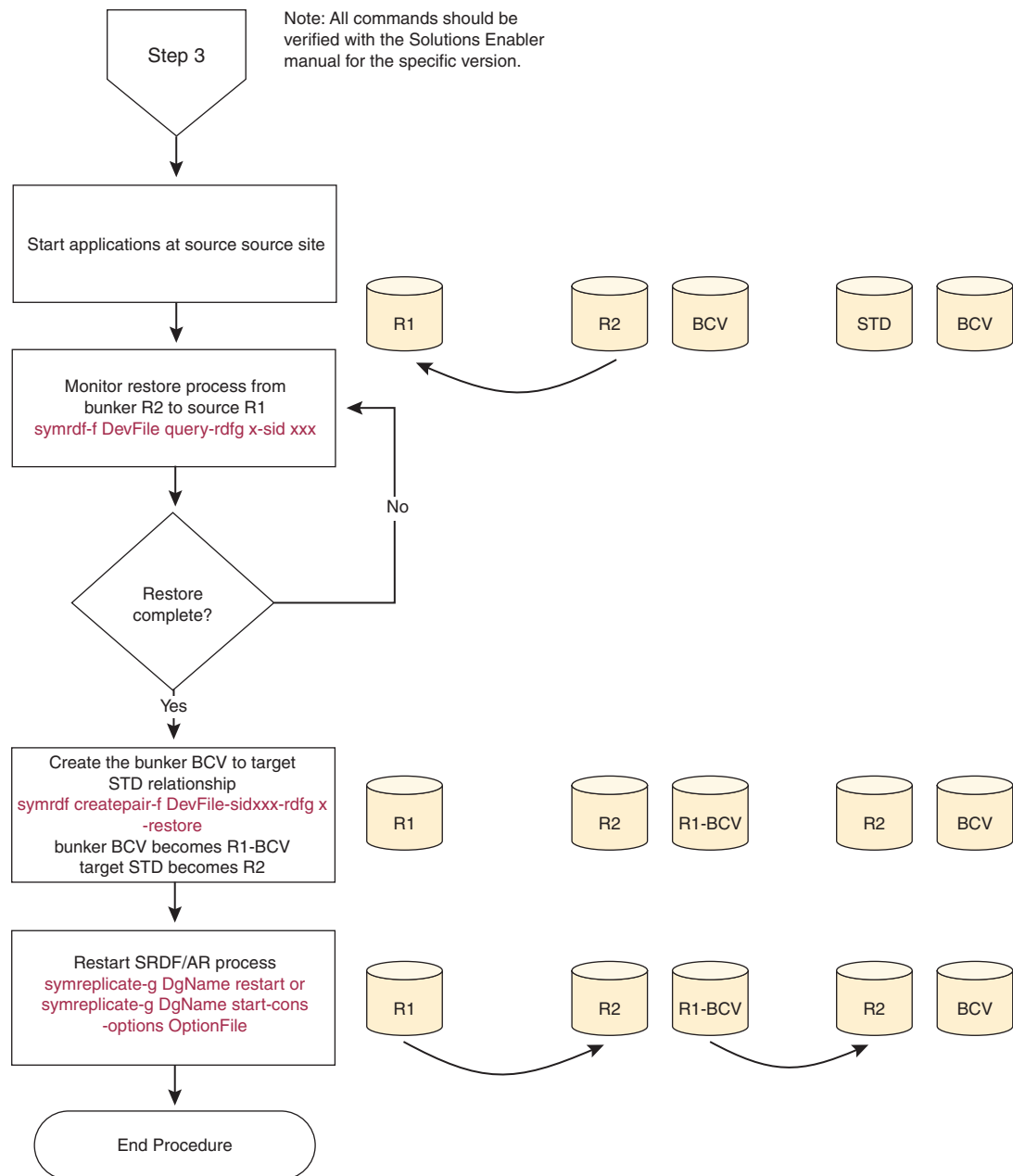


SYM-000626

Figure 7-15 Resynchronization Process

7.5.6.3 Step 3: Restart Applications, Reconfigure SRDF, and Restart SRDF/AR

Figure 7-16 demonstrates the restarting of the applications and reconfiguring the SRDF process required to restart SRDF/AR.



SYM-000627

Figure 7-16 Restart Applications, Reconfigure SRDF for SRDF/AR Restart

This appendix contains the following examples:

- ◆ A.1 Example 1: Creating an RDBMS-Restartable Copy on a Local Symmetrix.....A-2
- ◆ A.2 Example 2: Creating an RDBMS-Restartable BCV Copy on a Remote Symmetrix
Using PowerPath Support.....A-9
- ◆ A.3 Example 3: Performing Splits with Database AssistA-14

A.1 Example 1: Creating an RDBMS-Restartable Copy on a Local Symmetrix

The hardware setup for this example consists of an HP-UX production host (api183) and an HP-UX secondary host (api213). Both hosts are connected to a local Symmetrix (sid 814). PowerPath 2.1 is installed on api183. Oracle 8.1.7.0.0 is installed on api183 and api213. The example uses PowerPath devices 286–28B and BCV devices 39A–39F. An Oracle database was created on the production host.

All Oracle objects (data files, control files, and redo logs) must be on the PowerPath devices.

The example demonstrates consistent split operations both with and without PowerPath support.

The `sympd list` command issued from the production host displays a list of PowerPath devices on this Symmetrix that are visible to the host. The display below shows a portion of this list (PowerPath standard devices 286 through 28B). The ellipsis (...) represents truncated output.

Note that each PowerPath standard device has two SA (SCSI adapter) connections and physical device names (parent and child) that provide alternate paths for load-balancing and failover capabilities.

```
api183 # sympd list -powerpath
Symmetrix ID: 000185500814
```

P O W E R P A T H D E V I C E S								
Device Name	Directors			Device				
Physical	Sym	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)	
286			02B:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d2	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d2	-	04A:0	-	-	-	-	-	-
287			01B:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d3	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d3	-	04A:0	-	-	-	-	-	-
288			16A:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d4	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d4	-	04A:0	-	-	-	-	-	-
289			15A:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d5	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d5	-	04A:0	-	-	-	-	-	-
28A			02A:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d6	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d6	-	04A:0	-	-	-	-	-	-
28B			01A:C5	2-Way Mir	N/Grp'd	RW	4315	
/dev/rdisk/c0t1d7	-	13A:0	-	-	-	-	-	-
/dev/rdisk/c1t1d7	-	04A:0	-	-	-	-	-	-

The `sympd list` command issued from the secondary host (api213) displays all devices on the Symmetrix that are visible to the host. This partial list shows some devices that are configured as BCV devices. The example will use these BCVs to hold the DBMS-restartable copy of the database. To display just a list of BCVs on a Symmetrix, regardless of whether they are host-visible, use the `symdev list` command with the `-bcv` option.

```
api213 # sympd list
```

```
Symmetrix ID: 000185500814
```

Device Name	Directors				Device		
Physical	Sym	SA	:P DA	:IT Config	Attribute	Sts	Cap (MB)
/dev/rdisk/c0t0d0	39A	13B:0	16B:D5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d1	39B	13B:0	01A:D5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d2	39C	13B:0	16B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d3	39D	13B:0	15B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d4	39E	13B:0	02B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d5	39F	13B:0	01B:C5	BCV	N/Asst'd	RW	4315

Creating a device group and adding devices to it are prerequisites for performing the following operations. The `symsg create` command issued from the production host creates a device group named `oracle` on this host. The `syml addall` command adds the PowerPath standard devices to the group, using the command's `-range` option to limit the selections to those devices between 286 and 28B. The `symbcv` command associates the BCV devices with the device group; the `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
api183 # symsg create oracle
```

```
api183 # syml -g oracle addall dev -range 286:28B
```

```
api183 # symbcv -g oracle associateall dev -range 39A:39F
```

The `symmir establish` command initiates a full establish operation on all BCV pairs in the device group.

```
api183 # symmir -g oracle establish -full -noprompt
```

```
'Full Establish' operation execution is in progress for device group 'oracle'.
Please wait...
```

```
'Full Establish' operation successfully initiated for device group 'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends. The BCV pairs must be fully synchronized before you attempt a consistent split.

```
api183 # symmir -g oracle verify -i 30
```

```
None of the devices in group 'oracle' are in the Synchronized or Restored'
state.
```

```
Not all devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'oracle' are in the 'Synchronized or Restored' state.
```

When you issue the `symmir split` command with the `-rdb DbType` option and the database type is Oracle, SYMCLI utilizes the SYMAPI mapping features to automatically determine which PowerPath devices need to be frozen. SYMCLI logs into Oracle and queries Oracle's catalog. For SYMCLI to access Oracle, you need to set the environment variable `SYMCLI_RDB_CONNECT` to the username and password of the Oracle system administrator's

account. The following command sets the variable to a username of “system” and a password of “manager”.

```
api183 # export SYMCLI_RDB_CONNECT=system/manager
```

The export ORACLE_HOME command specifies the location of the Oracle binaries. The export ORACLE_SID command specifies the database instance name.

```
api183 # export ORACLE_HOME=/disks/symapidvt/oraclehome/api183
api183 # export ORACLE_SID=api183
```

The following command performs a consistent split using the `-rdb` option and specifies an Oracle database type. However, as the output from this command illustrates, the mapping facilities of SYMAPI determined that Oracle objects exist on only three of the six PowerPath devices in the device group. All six BCV pairs are split, but PowerPath freezes and thaws only those devices that contain the Oracle database (that is, the Oracle data files, control files and redo logs).

```
api183 # symmir -g oracle split -instant -rdb -dbtype oracle -noprompt
```

```
'Split' operation execution is in progress for device group 'oracle'.
Please wait...
```

```
Freezing 3 device(s).....Done.
Thawing 3 device(s).....Done.
```

```
'Split' operation successfully executed for device group 'oracle'.
```

A `symmir verify` command checks status of the background split every ten seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

```
None of the devices in group 'oracle' have finished splitting in the
background.
```

```
Not all devices in group 'oracle' have finished splitting in the
background.
```

```
All devices in group 'oracle' have finished splitting in the background.
```

The `symmir establish` command reestablishes incrementally the six BCV pairs in the device group that are currently in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'oracle'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

```
None of the devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

```
Not all devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'oracle' are in the 'Synchronized or Restored' state.
```

The following command performs a consistent split using the `-ppath` option and specifies the `stddevs` keyword to indicate that this operation applies to *all* PowerPath standard devices in the device group.

```
api183 # symmir -g oracle split -instant -ppath stddevs -noprompt
```

```
'Split' operation execution is in progress for device group 'oracle'.
Please wait...
```

```
Freezing 6 device(s).....Done.
Thawing 6 device(s).....Done.
'Split' operation successfully executed for device group 'oracle'.
```

The `symmir verify` command checks the status of the background split every ten seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

```
None of the devices in group 'oracle' have finished splitting in the
background.
```

```
Not all devices in group 'oracle' have finished splitting in the background.
```

```
All devices in group 'oracle' have finished splitting in the background.
```

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'oracle'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

```
None of the devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using the `-ppath` option and specifies explicitly the parent physical device name for each PowerPath device in the device group. However, EMC recommends using the `stddevs` keyword (as in the previous example) to specify all PowerPath standard devices in a device group.

```
api183 # symmir -g oracle split -instant -noprompt -ppath /dev/rdisk/c0t1d2 \  
         /dev/rdisk/c0t1d3 /dev/rdisk/c0t1d4 /dev/rdisk/c0t1d5 /dev/rdisk/c0t1d6 \  
         /dev/rdisk/c0t1d7
```

'Split' operation execution is in progress for device group 'oracle'.
Please wait...

```
      Freezing 6 device(s).....Done.  
      Thawing 6 device(s).....Done.
```

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are currently in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'oracle'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

```
None of the devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

```
Not all devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'oracle' are in the 'Synchronized or Restored' state.
```

The following command performs a consistent split using ECA (the `-consistent` option). This operation applies to all BCV pairs in the device group. When using ECA, you do not need PowerPath support. With ECA, the devices in the device group do not need to be PowerPath devices.

```
api183 # symmir -g oracle split -consistent -noprompt
```

```
'Split' operation execution is in progress for device group 'oracle'.
Please wait...
```

```
'Split' operation successfully executed for device group 'oracle'.
```

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

```
None of the devices in group 'oracle' have finished splitting in the
background.
```

```
Not all devices in group 'oracle' have finished splitting in the background.
```

```
All devices in group 'oracle' have finished splitting in the background.
```

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'oracle'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

```
None of the devices in group 'oracle' are in the 'Synchronized or Restored'
state.
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using the `-vxfs` option and specifies the names of four VERITAS file systems. When using VERITAS file systems, PowerPath is not required. The freeze and thaw tasks are carried out by the VERITAS file system.

```
api183 # symmir -g oracle split -instant -noprompt -vxfs /ora1 /ora2 /ora3 /ora4
```

'Split' operation execution is in progress for device group 'oracle'.
Please wait...

```
Freezing 4 filesystem(s).....Done.
Thawing 4 filesystem(s).....Done.
```

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

A.2 Example 2: Creating an RDBMS-Restartable BCV Copy on a Remote Symmetrix Using PowerPath Support

The hardware setup for this example consists of a Solaris production host (api179) and a Solaris secondary host (api180). The production host is connected to a local Symmetrix (sid 814); the secondary host is connected to a remote Symmetrix (sid 505). PowerPath 2.1 is installed on api179. Oracle 8.1.7.0.0 is installed on api179 and api180. The example uses local PowerPath standard devices 3C2–3C6 and remotely associated BCV devices 38A–38E.

An Oracle database was created on the production host. All Oracle objects (data files, control files and redo logs) must be on the PowerPath devices.

To perform a consistent split in an SRDF environment, you must run SRDF in synchronous mode.

All commands in this example are issued from the production host (api179).

The `symcfg list` command displays configuration information on the two Symmetrix units.

```
api179# symcfg list
```

```

                                S Y M M E T R I X

SymmID      Attachment  Model  Mcode Cache  Num Phys  Num Symm
              Version   Size (MB)  Devices    Devices
000185500814 Local      8530   5568    4088      450      1052
000185500505 Remote    8530   5568    8184       0       950

```

The `sympd list` command with the `-powerpath` option displays a list of PowerPath standard devices on the local Symmetrix that are visible to the host. The display below shows a portion of this list (Symmetrix devices 3C2 through 3C6). The ellipsis (...) represents truncated output.

```
api179# sympd list -powerpath
```

```
Symmetrix ID: 000185500814
```

```

                                P O W E R P A T H      D E V I C E S

Device Name      Directors      Device

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

3C2              16B:D5 RDF1              N/Grp'd      RW      4315
/dev/rdisk/c4t0d102s2 - 05B:0 - -              -            -            -
/dev/rdisk/c5t0d102s2 - 12B:0 - -              -            -            -

3C3              15B:D5 RDF1              N/Grp'd      RW      4315
/dev/rdisk/c4t0d103s2 - 05B:0 - -              -            -            -
/dev/rdisk/c5t0d103s2 - 12B:0 - -              -            -            -

3C4              02B:D5 RDF1              N/Grp'd      RW      4315
/dev/rdisk/c4t0d104s2 - 05B:0 - -              -            -            -
/dev/rdisk/c5t0d104s2 - 12B:0 - -              -            -            -

3C5              01B:D5 RDF1              N/Grp'd      RW      4315
/dev/rdisk/c4t0d105s2 - 05B:0 - -              -            -            -
/dev/rdisk/c5t0d105s2 - 12B:0 - -              -            -            -

3C6              16A:D5 RDF1              N/Grp'd      RW      4315
/dev/rdisk/c4t0d106s2 - 05B:0 - -              -            -            -
/dev/rdisk/c5t0d106s2 - 12B:0 - -              -            -            -

```

The `symdev list` command with the `-bcv` option displays those BCV devices on the remote Symmetrix (sid 505) that are not already part of a device group (N/Asst'd) and are free to be added to your device group. The display below has been edited to show only those devices that will be used in the example. Although these BCVs are not visible (accessible) to the production host, whose point of view is displayed, they are visible to the secondary host connected to the remote Symmetrix unit. The secondary host can access these BCVs and start the DBMS-restartable copy of the database that these BCVs will hold.

```
api179# symdev list -bcv -sid 505
```

```
Symmetrix ID: 000185500505
```

	Device Name	Directors	Device	Cap
Sym	Physical	SA :P DA :IT Config Attribute	Sts	(MB)
038A	Not Visible	???:? 02B:C5 BCV N/Asst'd	RW	4315
038B	Not Visible	???:? 01B:C5 BCV N/Asst'd	RW	4315
038C	Not Visible	???:? 16A:C5 BCV N/Asst'd	RW	4315
038D	Not Visible	???:? 15A:C5 BCV N/Asst'd	RW	4315
038E	Not Visible	???:? 02A:D4 BCV N/Asst'd	RW	4315

Creating a device group and adding devices to it are prerequisites for performing the following operations. The `symdg create` command issued from the production host creates an RDF1 type device group named `oracle` on this host. The `symld addall` command adds the PowerPath standard devices to the group, using the `-range` option to limit the selections to those devices between 286 and 28B. The `symbcv` command with the `-rdf` option associates the remote BCV devices with the device group; the command's `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
api179 # symdg create oracle -type rdf1
```

```
api179 # symld -g oracle addall dev -range 3C2:3C6
```

```
api179 # symbcv -g oracle associateall dev -range 38A:38E -rdf
```

The `symrdf establish` command initiates a full establish operation on all SRDF pairs in the device group. The local PowerPath standard (R1) devices will become synchronized with their respective target (R2) devices located on the remote Symmetrix.

```
api179# symrdf -g oracle establish nfull -noprompt
```

An RDF 'Full Establish' operation execution is in progress for device group 'oracle'. Please wait...

```
Suspend RDF link(s).....Done.
Mark target (R2) devices for full copy from source (R1)...Started.
Device: 3C2 ..... Marked.
Device: 3C3 ..... Marked.
Device: 3C4 ..... Marked.
Device: 3C5 ..... Marked.
Device: 3C6 ..... Marked.
Mark target (R2) devices for full copy from source (R1)...Done.
Merge device track tables between source and target.....Started.
Device: 3C2 ..... Merged.
Device: 3C3 ..... Merged.
Device: 3C4 ..... Merged.
Device: 3C5 ..... Merged.
Device: 3C6 ..... Merged.
Merge device track tables between source and target.....Done.
Read/Write Enable device(s) on SA at source (R1).....Done.
Resume RDF link(s).....Done.
```

The RDF 'Full Establish' operation successfully initiated for device group 'oracle'.

The `symrdf verify` command checks the state of the SRDF pairs in the device group every 30 seconds until the pairs are synchronized. Then the verify loop ends. The SRDF pairs must be fully synchronized before you attempt a consistent split.

```
api179# symrdf -g oracle verify -i 30
```

NONE 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 'oracle' are in the 'Synchronized' state.

The `symmir establish` command with the `-rdf` option initiates a full establish operation on the BCV pairs of the remote Symmetrix unit (sid 505). The BCV devices on that Symmetrix will become synchronized with the appropriate R2 devices on that Symmetrix.

```
api179# symmir -g oracle establish -full -rdf -noprompt
```

Remote 'Full Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Full Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command with the `-rdf` option checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized. The BCV pairs must be fully synchronized before you attempt a consistent split.

```
api179# symmir -g oracle verify -i 5 -rdf
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following `symmir` command performs a consistent split on the remote BCV pairs, using the command's `-ppath` option and specifying the `stddevs` keyword to indicate that this operation applies to *all* PowerPath standard devices in the device group.

```
api179# symmir -g oracle split -instant -rdf -ppath stddevs -noprompt
```

Remote 'Split' operation execution is in progress for device group 'oracle'. Please wait...

Freezing 5 device(s).....Done.

Thawing 5 device(s).....Done.

Remote 'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command with the `-rdf` option checks the status of the background split (`-bg`) of the remote BCV pairs every 10 seconds until it completes.

```
api179 # symmir -g oracle verify -split -bg -i 10 -rdf
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

The `symmir establish` command re-establishes incrementally the five remote BCV pairs in the device group that are currently in the Split state.

```
api179# symmir -g oracle establish -rdf -noprompt
```

Remote 'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

```
apil79# symmir -g oracle verify -i 5 -rdf
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

Some of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following `symmir` command performs a consistent split on the remote BCV pairs, using the command's `-ppath` option and specifying explicitly the parent physical device name for each PowerPath device in the device group. However, EMC recommends using the `stddevs` keyword (as in the previous example) to specify all PowerPath standard devices in a device group.

```
apil79# symmir -g oracle split -instant -rdf -noprompt -ppath \  
      /dev/rdisk/c4t0d102s2 /dev/rdisk/c4t0d103s2 /dev/rdisk/c4t0d104s2 \  
      /dev/rdisk/c4t0d105s2 /dev/rdisk/c4t0d106s2
```

Remote 'Split' operation execution is in progress for device group 'oracle'. Please wait...

```
Freezing 5 device(s).....Done.
```

```
Thawing 5 device(s).....Done.
```

Remote 'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split (`-bg`) of the remote BCV pairs every 10 seconds until it completes.

```
apil79 # symmir -g oracle verify -split -bg -i 10 -rdf
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

The `symmir establish` command re-establishes incrementally the five remote BCV pairs in the device group that are in the Split state.

```
apil79# symmir -g oracle establish -rdf -noprompt
```

Remote 'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

```
apil79# symmir -g oracle verify -i 5 -rdf
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

Some of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

A.3 Example 3: Performing Splits with Database Assist

Performing splits without PowerPath or ECA is similar for most supported databases in suspending writes to a database momentarily while an instant split occurs (refer to the section *A.3 Example 3: Performing Splits with Database Assist* on page A-14). The following examples show how to perform splits this way for an SQL 2000 database and an Oracle database.

A.3.1 Splitting BCV Pairs Using SQL 2000 Database Assist

This setup consists of a Windows 2000 host connected to a Symmetrix unit (sid 505). SQL Server 2000, Enterprise Edition, software is installed on the host. An SQL 2000 database called SQL2000a was created on Symmetrix devices. A device group called sqlserv was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

The `symmir query` command displays the configuration and status of the BCV pairs that hold the database. All BCV pairs are currently in the Synchronized state.

```
C:\> symmir -g sqlserv query
```

```
Device Group (DG) Name: sqlserv
DG's Type           : REGULAR
DG's Symmetrix ID    : 000185500505
```

Standard Device			BCV Device		State	
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. Tracks	STD <=> BCV
DEV001	0000	0	BCV001	00C8 *	0	Synchronized
DEV002	0001	0	BCV002	00C9 *	0	Synchronized
DEV003	0002	0	BCV003	00CA *	0	Synchronized
DEV004	0003	0	BCV004	00CB *	0	Synchronized
DEV005	0004	0	BCV005	00CC *	0	Synchronized
DEV006	0005	0	BCV006	00CD *	0	Synchronized
DEV007	0006	0	BCV007	00CE *	0	Synchronized
DEV008	0007	0	BCV008	00CF *	0	Synchronized
DEV009	0008	0	BCV009	00D0 *	0	Synchronized
DEV010	0009	0	BCV010	00D1 *	0	Synchronized
DEV011	000A	0	BCV011	00D2 *	0	Synchronized
DEV012	000B	0	BCV012	00D3 *	0	Synchronized
DEV013	000C	0	BCV013	00D4 *	0	Synchronized
DEV014	000D	0	BCV014	00D5 *	0	Synchronized
DEV015	000E	0	BCV015	00D6 *	0	Synchronized
DEV016	000F	0	BCV016	00D7 *	0	Synchronized
DEV017	0010	0	BCV017	00D8 *	0	Synchronized
DEV018	0011	0	BCV018	00D9 *	0	Synchronized
DEV019	0012	0	BCV019	00DA *	0	Synchronized
DEV020	0013	0	BCV020	00DB *	0	Synchronized
DEV021	0014	0	BCV021	00DC *	0	Synchronized
Total						
Track(s)		0			0	
MB(s)		0.0			0.0	

Legend:

(*): The paired BCV device is associated with this group.

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 `export` command sets this variable to a username of "system" and a password of "manager," allowing a local connection.

```
C:\> set SYMCLI_RDB_CONNECT=system/manager
```

The `symioctl begin snapshot` command sends the SQL command “BACKUP DATABASE TO VIRTUAL_DEVICE WITH SNAPSHOT” to SQL Server, which begins the snapshot backup and suspends writes to the database named SQL2000a. It also creates a savefile named `sql.save`.

```
C:\> symioctl begin snapshot SQL2000a savefile sql.save -type sqlserver -nop
```

The `symmir split` command performs an instant split on all BCV pairs in the device group `sqlserv`.

```
C:\> symmir -g sqlserv split -instant -noprompt
```

'Split' operation execution is in progress for device group 'sqlserv'.
Please wait...

'Split' operation successfully executed for device group 'sqlserv'.

The `symioctl end snapshot` command causes the SQL command “BACKUP DATABASE” to complete. Writes to the database can resume, and the snapshot metadata is saved to the save file.

```
C:\> symioctl end snapshot -type sqlserver SQL2000a -noprompt
```

A.3.2 Splitting BCV Pairs Using Oracle Database Assist

The hardware setup for this example consists of a Solaris host (`api179`) connected to a local Symmetrix unit. Oracle 8.1.7.0.0 software is installed on `api179`. A device group called `oracle` was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

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,” allowing a local connection to Oracle. 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
# export ORACLE_HOME=/disks/symapidvt/oraclehome/api179
# export ORACLE_SID=api179
```

The `symrdb list` command allows you to test basic database connectivity.

```
# symrdb list -type oracle
```

```
DATABASE NAMES (ORACLE 8.1.7.0.0):
```

```
Database Name
```

```
-----
```

```
api179
```

The `symioctl begin backup` command places all tables in the Oracle database (`api179`) in hot backup mode.

```
# symioctl -type oracle begin backup -noprompt
```

The `symioctl freeze` command suspends writes to the Oracle database (`api179`).

```
# symioctl -type oracle freeze -noprompt
```

The `symmir split` command performs an instant split on all BCV pairs in the device group called `oracle`.

```
# symmir -g oracle split -instant -noprompt
```

```
'Split' operation execution is in progress for device group 'oracle'.  
Please wait...
```

```
'Split' operation successfully executed for device group 'oracle'.
```

The `symioctl thaw` command allows writes to the database to resume.

```
# symioctl -type oracle thaw -noprompt
```

The `symioctl end backup` command terminates hot backup mode.

```
# symioctl -type oracle end backup -noprompt
```

This appendix provides the following examples:

- ◆ B.1 Example 1: Single-Hop Configuration B-2
- ◆ B.2 Example 2: Multihop Configuration with BCVs at Hop 2 B-14
- ◆ B.3 Example 3: MultiHop Configuration without BCVs at Hop 2 B-22
- ◆ B.4 Example 4: Accessing Concurrent Non-SRDF/AR BCVs while Running SRDF/AR
(Single-Hop Configuration) B-31
- ◆ B.5 Example 5: Accessing Concurrent Non-SRDF/AR BCVs while Running SRDF/AR
(Multihop Configuration)..... B-35
- ◆ B.6 Example 6: Restarting a Replicate Session when Devices Are Locked B-40

B.1 Example 1: Single-Hop Configuration

The hardware setup for this example consists of an 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 and shows how to set up the correct pair states for automated data replication in this environment:

Local Source Symmetrix (sid 161): standard devices 27C-285; R1-BCV devices 2E0-2E9

Remote Target Symmetrix (sid 505): R2 devices 27C-285, BCV devices 2E0-2E9

```
# symcfg list
```

```

                                S Y M M E T R I X

SymmID      Attachment  Model  Mcode  Cache  Num Phys  Num Symm
              Version   Size (MB)  Devices  Devices
000184501161 Local        8430  5567   16384    545     967
000185500505 Remote       8430  5567    8192     0     912

```

Creating a device group and adding devices to it are prerequisites for performing Symmetrix Automated Replication. The `syndg create` command creates a device group named `symrep`.

```
# syndg create symrep
```

The `sympd list` command displays all Symmetrix devices that are visible to this 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 device group `symrep`.

```
# sympd list
```

Symmetrix ID: 000184501161

Device Name	Directors	Device	Cap (MB)
Physical	Sym SA :P DA :IT Config	Attribute	Sts
/dev/rdsk/c1t0d0	27C 04A:0 16B:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d1	27D 04A:0 15B:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d2	27E 04A:0 02B:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d3	27F 04A:0 01B:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d4	280 04A:0 16A:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d5	281 04A:0 15A:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d6	282 04A:0 02A:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t0d7	283 04A:0 01A:D5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t1d0	284 04A:0 16B:C5 2-Way Mir	N/Grp'd	RW 4315
/dev/rdsk/c1t1d1	285 04A:0 15B:C5 2-Way Mir	N/Grp'd	RW 4315

The `symlld` 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 27C through 285).

```
# symlld -g symrep addall dev -range 27C:285
```

The `symdev list` command with the `-r1 -bcv` options displays those R1 BCV devices in the local Symmetrix system (sid 161) 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 only those devices that will be used in the example.

```
# symdev list -r1 -bcv
```

Symmetrix ID: 000184501161

Device Name	Directors	Device			
Sym Physical	SA :P DA :IT Config	Attribute	Sts	Cap (MB)	
2E0 Not Visible	??? :? 15A:D5 RDF1-BCV	N/Asst'd	RW	4315	
2E1 Not Visible	??? :? 02A:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E2 Not Visible	??? :? 02A:D5 RDF1-BCV	N/Asst'd	RW	4315	
2E3 Not Visible	??? :? 15A:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E4 Not Visible	??? :? 01A:D5 RDF1-BCV	N/Asst'd	RW	4315	
2E5 Not Visible	??? :? 16A:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E6 Not Visible	??? :? 16B:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E7 Not Visible	??? :? 01B:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E8 Not Visible	??? :? 15B:C5 RDF1-BCV	N/Asst'd	RW	4315	
2E9 Not Visible	??? :? 02B:C5 RDF1-BCV	N/Asst'd	RW	4315	

The `symbcv` command associates the local BCV devices with the device group; the `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
# symbcv -g symrep associateall dev -range 2E0:2E9
```

The `symdev list` command with the `-bcv` option displays those BCV devices in the remote Symmetrix (sid 505) 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 only those devices that will be used in the example.

```
# symdev list -sid 0505 -bcv
```

Symmetrix ID: 000185500505

Device Name	Directors	Device			
Sym Physical	SA :P DA :IT Config	Attribute	Sts	Cap (MB)	
2E0 Not Visible	??? :? 16B:D5 BCV	N/Asst'd	RW	4315	
2E1 Not Visible	??? :? 15B:D5 BCV	N/Asst'd	RW	4315	
2E2 Not Visible	??? :? 02B:D5 BCV	N/Asst'd	RW	4315	
2E3 Not Visible	??? :? 15A:C0 BCV	N/Asst'd	RW	4315	
2E4 Not Visible	??? :? 01B:D5 BCV	N/Asst'd	RW	4315	
2E5 Not Visible	??? :? 16A:C0 BCV	N/Asst'd	RW	4315	
2E6 Not Visible	??? :? 16A:D5 BCV	N/Asst'd	RW	4315	
2E7 Not Visible	??? :? 15A:D5 BCV	N/Asst'd	RW	4315	
2E8 Not Visible	??? :? 02A:D5 BCV	N/Asst'd	RW	4315	
2E9 Not Visible	??? :? 01A:D5 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 -rdf -bcv associateall dev -range 2E0:2E9
```

The `symdg show` command displays detailed group information about device group `symrep`. The group contains 10 local standard devices, 10 local R1 BCVs, and 10 remote BCVs.

```
# symdg show symrep
```

Group Name: `symrep`

```
Group Type           : REGULAR
Valid                : Yes
Symmetrix ID         : 000184501161
Group Creation Time   : Tue Jun 12 10:11:52 2001
Vendor ID            : EMC Corp
Application ID        : SYMCLI
```

```
Number of STD Devices in Group : 10
Number of Associated GK's       : 0
Number of Locally-associated BCV's : 10
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 10
```

Standard (STD) Devices (10):
{

LdevName	PdevName	Sym Dev	Att. Sts	Cap (MB)
DEV001	/dev/rdisk/clt0d0	27C	RW	4315
DEV002	/dev/rdisk/clt0d1	27D	RW	4315
DEV003	/dev/rdisk/clt0d2	27E	RW	4315
DEV004	/dev/rdisk/clt0d3	27F	RW	4315
DEV005	/dev/rdisk/clt0d4	280	RW	4315
DEV006	/dev/rdisk/clt0d5	281	RW	4315
DEV007	/dev/rdisk/clt0d6	282	RW	4315
DEV008	/dev/rdisk/clt0d7	283	RW	4315
DEV009	/dev/rdisk/clt1d0	284	RW	4315
DEV010	/dev/rdisk/clt1d1	285	RW	4315

BCV Devices Locally-associated (10):
{

LdevName	PdevName	Sym Dev	Att. Sts	Cap (MB)
BCV001	N/A	2E0	RW	4315
BCV002	N/A	2E1	RW	4315
BCV003	N/A	2E2	RW	4315
BCV004	N/A	2E3	RW	4315
BCV005	N/A	2E4	RW	4315
BCV006	N/A	2E5	RW	4315
BCV007	N/A	2E6	RW	4315
BCV008	N/A	2E7	RW	4315
BCV009	N/A	2E8	RW	4315
BCV010	N/A	2E9	RW	4315

BCV Devices Remotely-associated (BCV RDF) (10):

```
{
-----

```

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
BRBCV001	N/A	2E0		RW	4315
BRBCV002	N/A	2E1		RW	4315
BRBCV003	N/A	2E2		RW	4315
BRBCV004	N/A	2E3		RW	4315
BRBCV005	N/A	2E4		RW	4315
BRBCV006	N/A	2E5		RW	4315
BRBCV007	N/A	2E6		RW	4315
BRBCV008	N/A	2E7		RW	4315
BRBCV009	N/A	2E8		RW	4315
BRBCV010	N/A	2E9		RW	4315

```
}
-----
```

Device Group BCV RDF Information

```
{
RDF Type : R1
RDF (RA) Group Number : 2 (B)

Remote Symmetrix ID : 000185500505

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 : Direct Escon 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

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 : 1840
}
```

The `symmir establish` command establishes the local BCV pairs. A side effect of this command is to suspend the related SRDF pairs (local R1 BCVs with their R2 mirrors) if the SRDF pairs are still synchronized at the time.

```
# symmir -g symrep establish -full -noprompt
```

```
'Full Establish' operation execution is in progress for device group
'symrep'. Please wait...
```

```
'Full Establish' operation successfully initiated for device group
'symrep'.
```

The `symmir verify` command checks the state of the local BCV pairs in the device group every 30 seconds until those BCV pairs are synchronized. Then the verify loop ends.

```
# symmir -g symrep verify -i 30
```

```
None of the devices in group 'symrep' are in the 'Synchronized or Restored'
state.
```

```
Not all devices in group 'symrep' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'symrep' are in the 'Synchronized or Restored' state.
```

The `symmir split` command with the `-instant` option initiates an instant split on the local BCV pairs. When the background split completes, the RDF links for the R1 BCV devices will still be not ready (NR), even though the state of the R1 BCV devices will be changed from not ready to read/write enabled. For more information on splits, refer to the white paper *Using SYMCLI to Perform TimeFinder Control Operations* (P/N 300-000-074).

```
# symmir -g symrep split -instant -noprompt
```

```
'Split' operation execution is in progress for device group 'symrep'.
Please wait...
```

```
'Split' operation successfully executed for device group 'symrep'.
```

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
# symmir -g symrep verify -split -bg -i 10
```

```
None of the devices in group 'symrep' have finished splitting in the
background.
```

```
Not all devices in group 'symrep' have finished splitting in the
background.
```

```
All devices in group 'symrep' have finished splitting in the background.
```

The `symrdf establish` command with the `-bcv` option resumes the RDF links for the R1 BCV devices and initiates the propagation of data from the local R1 BCV devices to their remote R2 mirror devices.

```
# symrdf -g symrep establish -bcv -noprompt
```

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

```
Suspend RDFlink(s).....Done.
```

```
Resume RDF link(s).....Done.
```

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

The `symrdf query` command shows again the SRDF pair state of the R1 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 symrep query -bcv
```

```
Device Group (DG) Name : symrep
DG's Type              : REGULAR
DG's Symmetrix ID      : 000184501161
```

R E M O T E S Y M M E T R I X									
Source (R1) View					Target (R2) View				
-----					-----				
BCV	ST				LI	ST			M
Logical	A				N	A			o
Device	T R1 Inv	R2 Inv			K	T R1 Inv	R2 Inv		d
Dev	E Tracks	Tracks			S Dev	E Tracks	Tracks		e
-----					-----				
BCV001	2E0 RW	0	417 RW	27C WD	0	0 SYN	DIS	OFF	SyncInProg
BCV002	2E1 RW	0	401 RW	27D WD	0	0 SYN	DIS	OFF	SyncInProg
BCV003	2E2 RW	0	385 RW	27E WD	0	0 SYN	DIS	OFF	SyncInProg
BCV004	2E3 RW	0	417 RW	27F WD	0	0 SYN	DIS	OFF	SyncInProg
BCV005	2E4 RW	0	385 RW	280 WD	0	0 SYN	DIS	OFF	SyncInProg
BCV006	2E5 RW	0	58 RW	281 WD	0	0 SYN	DIS	OFF	SyncInProg
BCV007	2E6 RW	0	54 RW	282 WD	0	0 SYN	DIS	OFF	SyncInProg
BCV008	2E7 RW	0	51 RW	283 WD	0	0 SYN	DIS	OFF	SyncInProg
BCV009	2E8 RW	0	52 RW	284 WD	0	0 SYN	DIS	OFF	SyncInProg
BCV010	2E9 RW	0	37 RW	285 WD	0	0 SYN	DIS	OFF	SyncInProg

Total									
Track(s)	0	2257			0	0			
MB(s)	0.0	70.0			0.0	0.0			

The `symmir establish` command establishes the remote BCV pairs.

```
# symmir -g symrep establish -bcv -rdf -full -noprompt
```

```
Remote 'Full Establish' operation execution is in progress for device group
'symrep'. Please wait...
```

```
Remote 'Full Establish' operation successfully initiated for device group
'symrep'.
```

The `symmir query` command checks the state of the remote BCV pairs. They are still in the process of synchronizing.

```
# symmir -g symrep query -bcv -rdf
```

```
Device Group (DG) Name: symrep
DG's Type               : REGULAR
DG's Symmetrix ID       : 000184501161
Remote Symmetrix ID     : 000185500505
```

```

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

Standard Device                BCV Device                State
-----
Logical      Inv.              Logical              Inv.              State
Sym Tracks   Sym Tracks   Sym Tracks   STD <=> BCV
-----
BCV001       27C             0 BRBCV001   2E0 * 137073 SyncInProg
BCV002       27D             0 BRBCV002   2E1 * 137433 SyncInProg
BCV003       27E             0 BRBCV003   2E2 * 137153 SyncInProg
BCV004       27F             0 BRBCV004   2E3 * 134431 SyncInProg
BCV005       280             0 BRBCV005   2E4 * 135924 SyncInProg
BCV006       281             0 BRBCV006   2E5 * 137696 SyncInProg
BCV007       282             0 BRBCV007   2E6 * 138090 SyncInProg
BCV008       283             0 BRBCV008   2E7 * 134496 SyncInProg
BCV009       284             0 BRBCV009   2E8 * 138068 SyncInProg
BCV010       285             0 BRBCV010   2E9 * 138090 SyncInProg

```

The `symmir verify` command checks the state of the remote BCV pairs in the device group every 30 seconds until those BCV pairs are synchronized. Then the verify loop ends.

```
# symmir -g symrep -bcv -rdf verify -i 30
```

```
None of the devices in group 'symrep' are in the 'Synchronized or Restored'
state.
```

```
Not all devices in group 'symrep' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'symrep' are in the 'Synchronized or Restored' state.
```

The `symmir split` command performs an instant split on the remote BCV pairs in the device group.

```
# symmir -g symrep split -instant -bcv -rdf -noprompt
```

```
Remote 'Split' operation execution is in progress for device group
'symrep'. Please wait...
```

```
Remote 'Split' operation successfully executed for device group 'symrep'.
```

The `symmir verify` command with the `-bcv -rdf` and `-split -bg` options checks the background split state of the remote BCV pairs in the device group every 10 seconds until the background split is complete for those pairs.

```
# symmir -g symrep -bcv -rdf verify -split -bg -i 10
```

```
None of the devices in group 'symrep' have finished splitting in the
background.
```

```
Not all devices in group 'symrep' have finished splitting in the
background.
```

```
All devices in group 'symrep' have finished splitting in the background.
```

The `symmir establish` command re-establishes the local BCV pairs. As a result, the RDF links between the local BCVs and their remote mirrors are automatically suspended.

```
# symmir -g symrep establish -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'symrep'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'symrep'.
```

The requirements for beginning a replicate session in a single-hop configuration have now been met:

- ◆ Local BCV pairs are established.
- ◆ SRDF pairs are suspended.
- ◆ Remote BCV pairs are split.

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 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.

```
# symreplicate -g symrep start -options symrep.opt -foreground -noprompt
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Checking for local BCV synchronization...
```

```
Splitting local BCV pairs...
```

```
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...
```

This `symreplicate` command runs one copy cycle using the `-ppath stddevs` option to freeze I/O to all local standard PowerPath devices in the device group (in this case, the 10 standard devices that belong to device group `symrep`). PowerPath suspends I/O to these devices before splitting the local BCV pairs. I/O resumes when the foreground split completes. Suspending database I/O during a TimeFinder instant split results in a DBMS restartable copy of the database on the remote Symmetrix.

```
# symreplicate -g symrep start -options symrep.opt -foreground -noprompt \
  -ppath stddevs
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Checking for local BCV synchronization...
```

```
Splitting local BCV pairs...
```

```
Freezing 10 device(s).....Done.
```

```
Thawing 10 device(s).....Done.
```

```
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...
```

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          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Active	Waiting for next cycle		10 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Active	Splitting local BCV pairs		10 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Active	Establishing RDF pairs		10 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Active	Establishing local BCV pairs		10 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Active	Establishing remote BCV pairs		10 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate				Cycle	Current	Max
Hop	Type	Status	Step	Period	Cycle	Cycles
SINGLE	Completed	Complete		10 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          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max 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 after establishing the remote BCV pairs.

```
# symreplicate -g symrep query
```

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Stopped	Establishing remote BCV pairs	10 m	1	2

The `symreplicate restart` command resumes the copy cycle at the step where the session stopped.

```
# symreplicate -g symrep restart -options symrep.opt -noprompt
```

Checking for valid group configuration...

`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          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Waiting for next cycle		10 m	1	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Splitting local BCV pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing RDF pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing local BCV pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing local BCV pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing remote BCV pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000184501161
Remote Symmetrix ID        : 000185500505
```

Replicate Hop	Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Completed	Complete		10 m	2	2

B.2 Example 2: Multihop Configuration with BCVs at Hop 2

The hardware setup for this example consists of a Solaris host connected to a source Symmetrix (sid 232). All commands are issued from the source-side host. The example uses the following devices to create the multihop environment and shows how to set up the correct pair states for automated data replication in this environment:

Source Symmetrix (sid 232): R1 devices 002-007

Hop 1 Symmetrix (sid 231): R2 devices 000-005; R1 BCV devices 048-04D

Hop 2 Symmetrix (sid 233): R2 devices 000-005; BCV devices 030-035

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 232
```

```
Symmetrix ID: 000000005232
```

Local Device View												
		STATUS				M O D E S			RDF S T A T E S			
Sym Dev	RDev	RDF Typ:G	SA	RA	LNK	Mode	Dom	ACp	R1 Inv Tracks	R2 Inv Tracks	Dev	RDev Pair
002	000	R1:1	RW	RW	NR	SYN	DIS	OFF	0	1	RW	RW Split
003	001	R1:1	RW	RW	NR	SYN	DIS	OFF	0	1	RW	RW Split
004	002	R1:1	RW	RW	NR	SYN	DIS	OFF	0	1	RW	RW Split
005	003	R1:1	RW	RW	NR	SYN	DIS	OFF	0	1	RW	RW Split
006	004	R1:1	RW	RW	NR	SYN	DIS	OFF	0	0	RW	RW Split
007	005	R1:1	RW	RW	NR	SYN	DIS	OFF	0	0	RW	RW Split
008	006	R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD Synchronized
009	007	R1:1	RW	RW	RW	SYN	DIS	OFF	-	-	RW	WD Synchronized
00A	008	R1:1	RW	RW	RW	SYN	DIS	OFF	-	-	RW	WD Synchronized
Total									0	96		
Track(s)									0.0	3.0		
MB(s)												

Creating a device group and adding devices to it are prerequisites for performing Symmetrix Automated Replication. The `symdbg create` command creates a device group (symrep). The `symld addall` command adds the RDF standard devices from local Symmetrix 000000005232 to the group, using the `-range` option to limit the selections to those devices between 002 and 007. The `symbcv` command with the `-rdf` option associates remote BCV devices on Hop 1 with the device group; the `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
# symdbg create symrep -type rdf1
# symld -g symrep addall dev -range 002:007 -sid 232
# symbcv -g symrep -rdf associateall dev -range 48:4D
```

The local RDF standard devices chosen for the example are in the Split state. The following `symrdf establish` command initiates the synchronization of these SRDF pairs.

```
# symrdf -g symrep establish -noprompt
```

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

```
Write Disable device(s) on RA at target (R2).....Done.
Suspend RDF link(s).....Done.
Resume RDF link(s).....Not Done.
Merge device track tables between source and target.....Started.
Devices: 002-007 ..... 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 'symrep'.

The following query indicates that these local SRDF pairs are now synchronized:

```
# symrdf -g symrep query
```

```
Device Group (DG) Name: symrep
DG's Type           : RDF1
DG's Symmetrix ID    : 000000005232
```

Source (R1) View					Target (R2) View				M O D E S				
-----					-----				-----				
Standard	ST				LI	ST				M			
Logical	A				N	A				O			
Device	T R1 Inv	R2 Inv			K	T R1 Inv	R2 Inv			d			
Dev	E Tracks	Tracks			S Dev	E Tracks	Tracks		e	Dom	ACp	STATE	
-----					-----				-----				
DEV001	002 RW	0	0	0	RW 000	WD	0	0	0	SYN	DIS	OFF	Synchronized
DEV002	003 RW	0	0	0	RW 001	WD	0	0	0	SYN	DIS	OFF	Synchronized
DEV003	004 RW	0	0	0	RW 002	WD	0	0	0	SYN	DIS	OFF	Synchronized
DEV004	005 RW	0	0	0	RW 003	WD	0	0	0	SYN	DIS	OFF	Synchronized
DEV005	006 RW	0	0	0	RW 004	WD	0	0	0	SYN	DIS	OFF	Synchronized
DEV006	007 RW	0	0	0	RW 005	WD	0	0	0	SYN	DIS	OFF	Synchronized
Total													
Track(s)	0	0				0	0						
MB(s)	0.0	0.0				0.0	0.0						

The `symmir establish` command initiates the creation of remote BCV pairs on Hop 1. Using the `-rdf` option targets the remote BCVs in the device group.

```
# symmir -g symrep establish -full -rdf -noprompt
```

Remote 'Full Establish' operation execution is in progress for device group 'symrep'. Please wait...

Remote 'Full Establish' operation successfully initiated for device group 'symrep'.

The `symmir query` command displays the remote BCV pairs on Hop 1 and their current state (SyncInProgress).

```
# symmir -g symrep -rdf query
```

```
Device Group (DG) Name: symrep
DG's Type           : RDF1
DG's Symmetrix ID   : 000000005232
Remote Symmetrix ID : 000000005231
```

```

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

Standard Device                BCV Device                State
-----
Logical      Inv.              Logical      Inv.              STD <=> BCV
              Sym Tracks
-----
DEV001        000          0 RBCV001        048 * 10212 SyncInProgress
DEV002        001          0 RBCV002        049 * 16110 SyncInProgress
DEV003        002          0 RBCV003        04A * 16955 SyncInProgress
DEV004        003          0 RBCV004        04B * 17915 SyncInProgress
DEV005        004          0 RBCV005        04C * 20418 SyncInProgress
DEV006        005          0 RBCV006        04D * 21710 SyncInProgress

Total          -----
Track(s)              0          103320
MB(s)                0.0          3228.8

```

Legend:

(*): The paired BCV device is associated with this group.

The `symmir verify` command with the `-rdf` option checks the state of the Hop 1 BCV pairs every 30 seconds until those BCV pairs are synchronized. Then the verify loop ends.

```
# symmir -g symrep verify -rdf -i 30
```

None of the devices in group 'symrep' are in the 'Synchronized or Restored' state.

Not all devices in group 'symrep' are in the 'Synchronized or Restored' state.

All devices in group 'symrep' are in the 'Synchronized or Restored' state.

While part of a BCV pair, the R1 BCVs on Hop 1 are temporarily suspended from copying data to their remote R2 targets. To copy data from these BCVs to their respective Hop 2 target devices, the example performs a TimeFinder instant split on the BCV pairs on Hop 1 and subsequently re-establishes the BCVs with their Hop 2 target devices.

```
# symmir -g symrep split -instant -rdf -noprompt
```

```
Remote 'Split' operation execution is in progress for device group
'symrep'. Please wait...
```

```
Remote 'Split' operation successfully executed for device group 'symrep'.
```

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
# symmir -g symrep verify -split -rdf -bg -i 10
```

None of the devices in group 'symrep' have finished splitting in the background.

Not all devices in group 'symrep' have finished splitting in the background.

All devices in group 'symrep' have finished splitting in the background.

The `symrdf establish` command resumes the RDF links for the R1 BCV devices on Hop 1 and initiates the propagation of data from these BCVs to their remote R2 target devices on Hop 2.

```
# symrdf -g symrep establish -rbcv -noprompt
```

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

```
Suspend RDF link(s).....Done.
Resume RDF link(s).....Done.
```

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

The `symrdf query` command displays the remote SRDF pairs and their current state (SyncInProg).

```
# symrdf -g symrep query -rbcv
```

```
Device Group (DG) Name: symrep
DG's Type           : RDF1
DG's Symmetrix ID   : 000000005232
RBCV's Symmetrix ID : 000000005231
```

R E M O T E												S Y M M E T R I X											
Source (R1) View						Target (R2) View						M O D E S											
-----						-----						-----											
		ST		LI		ST		M															
BCV	A				N	A			O														
Logical	T	R1 Inv	R2 Inv	K	T	R1 Inv	R2 Inv	d				RDF	Pair										
Device	Dev	E Tracks	Tracks	S	Dev	E Tracks	Tracks	e	Dom	ACp	STATE												

RBCV001	048	RW	0	26187	RW	000	WD	0	0	SYN	DIS	OFF	SyncInProg										
RBCV002	049	RW	0	26737	RW	001	WD	0	0	SYN	DIS	OFF	SyncInProg										
RBCV003	04A	RW	0	23730	RW	002	WD	0	0	SYN	DIS	OFF	SyncInProg										
RBCV004	04B	RW	0	22320	RW	003	WD	0	0	SYN	DIS	OFF	SyncInProg										
RBCV005	04C	RW	0	23193	RW	004	WD	0	0	SYN	DIS	OFF	SyncInProg										
RBCV006	04D	RW	0	23590	RW	005	WD	0	0	SYN	DIS	OFF	SyncInProg										
Total																							
Track(s)		-----		-----		-----		-----															
		0		145757		0		0															
MB(s)		0.0		4554.0		0.0		0.0															

The `symrdf verify` command with the `-rbcv` option displays a message every 30 seconds until all of the remote SRDF pairs (Hop 1 BCVs and their remote R2 target devices) are synchronized.

```
# symrdf -g symrep verify -rbcv -i 30
```

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

The following command illustrates the use of the `vi` text editor to create a text file¹ named `lasthop` and enter the BCV pairs that will be established on the Hop 2 Symmetrix system. The file lists each BCV pair on a separate line with the R2 device name first and the BCV device name second (the R2 device 000 with BCV 030, the R2 device 001 with BCV 031, and so forth).

```
# vi lasthop
```

```
000 030
001 031
002 032
003 033
004 034
005 035
```

The `symmir establish` command initiates synchronization of the BCV pairs at the Hop 2 site. The `-f` option applies the text file, `lasthop`, that lists each BCV pair. The `-sid` option applies the command to the remote Symmetrix unit (000000005233). These two options allow you to operate on remote BCV pairs in the Hop 2 Symmetrix of a multihop configuration.

```
# symmir -f lasthop -sid 33 establish -full -noprompt
```

```
'Full Establish' operation execution is in progress for the device list in
device file 'lasthop'. Please wait...
```

```
'Full Establish' operation successfully initiated for the device list in
device file 'lasthop'.
```

-
1. If you are using Solutions Enabler version 5.3 or higher, you can include the Hop 2 BCVs in the device group (instead of creating a device file for them as shown here). For example:

```
symbcv -g symrep associateall -range 30:35 -rrdf
```

All subsequent commands involving the device file (`-f lasthop`) can then be performed on the device group (`-g symrep`) using the option `-rbcv`, which targets the Hop 2 BCV devices. For example:

```
symmir -g symrep -sid 33 establish -full -noprompt -rbcv
```

The `symmir query` command displays the remote BCV pairs on the Hop 2 Symmetrix system and their current pair state (SyncInProgress).

```
# symmir -f lasthop -sid 33 query -noprompt
```

```
Device File Name      : lasthop
Device's Symmetrix ID : 000000005233
```

Standard Device			BCV Device		State	
Logical	Inv. Sym Tracks	Logical	Sym	Inv. Tracks	STD	<=> BCV
N/A	000	0 N/A	030	31502	SyncInProg	
N/A	001	0 N/A	031	29940	SyncInProg	
N/A	002	0 N/A	032	30746	SyncInProg	
N/A	003	0 N/A	033	32455	SyncInProg	
N/A	004	0 N/A	034	31325	SyncInProg	
N/A	005	0 N/A	035	31641	SyncInProg	
Total	-----			-----		
Track(s)		0		187609		
MB(s)		0.0		5862.8		

Legend:

(*): The paired BCV device is associated with this group.

The `symmir verify` command checks the state of the Hop 2 BCV pairs every 30 seconds until these BCV pairs are synchronized.

```
# symmir -f lasthop -sid 33 verify -i 30
```

None of the devices in the list are in the 'Synchronized or Restored' state.

Not all devices in the list are in the 'Synchronized or Restored' state.

All devices in the list are in the 'Synchronized or Restored' state.

Initial setup for using Symmetrix Automated Replication requires that the BCV pairs on Hop 2 be split. The following `symmir split` command initiates an instant split on the Hop 2 BCV pairs.

```
# symmir -f lasthop -sid 33 split -instant -noprompt
```

```
'Split' operation execution is in progress for the device list in device file
'lasthop'. Please wait...
```

```
'Split' operation successfully executed for the device list in device file
'lasthop'.
```

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
# symmir -f lasthop verify -split -bg -i 10
```

None of the devices in the list have finished splitting in the background.

Not all devices in the list have finished splitting in the background.

All devices in the list have finished splitting in the background.

Initial setup for using Symmetrix Automated Replication also requires that the remote BCV pairs on Hop 1 be synchronized and that the remote SRDF pairs be suspended. Using the `symmir establish` command with the `-rdf` option re-establishes the remote BCV pairs and, at the same time, suspends the remote SRDF pairs (the Hop 1 BCVs with their R2 target devices on Hop 2).

```
# symmir -g symrep establish -rdf -noprompt
```

Remote 'Incremental Establish' operation execution is in progress for device group 'symrep'. Please wait...

Remote 'Incremental Establish' operation successfully initiated for device group 'symrep'.

The `symmir verify` command with the `-rdf` option checks the state of the Hop 1 BCV pairs every 30 seconds until these remote BCV pairs are synchronized. When the verify loop ends, initial setup for Symmetrix Automated Replication is complete for the multihop configuration with BCVs at Hop 2.

```
# symmir -g symrep verify -rdf -i 30
```

None of the devices in group 'symrep' are in the 'Synchronized or Restored' state.

Not all devices in group 'symrep' are in the 'Synchronized or Restored' state.

All devices in group 'symrep' are in the 'Synchronized or Restored' state.

The requirements for beginning a replicate session in a multihop configuration with BCVs at Hop 2 have now been met:

- ◆ Local SRDF pairs are synchronized.
- ◆ BCV pairs on Hop 1 are synchronized.
- ◆ Remote SRDF pairs are suspended.
- ◆ BCV pairs on Hop 2 are split.

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 multihop configuration that uses the Hop 2 BCVs (`USE_FINAL_BCV=TRUE`) 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 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=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=1
```

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 specified file, `rep_opt.txt`, tells `SYMCLI` what copy cycle parameters to employ during the session.

```
# symreplicate -g symrep start -options rep_opt.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...
1 cycle(s) complete; exiting symreplicate...
```

B.3 Example 3: MultiHop Configuration without BCVs at Hop 2

The hardware setup for this example consists of a Solaris host connected to a source Symmetrix (sid 787). All commands are issued from the source-side host. The following devices are used to create the multihop environment and set up the correct pair states for automated data replication in this environment:

Source Symmetrix (sid 787): R1 devices 01F-022

Hop 1 Symmetrix (sid 549): R2 devices 021-024; R1 BCV devices 089, 090, 091 and 092

Hop 2 Symmetrix (sid 601): R2 devices 087, 08E, 08F, and 090

The `symrdf list` command shows the local view of the source R1 devices (SymDev), the target R2 mirror devices (RDev), and their current SRDF pair state (Synchronized). The ellipsis (.....) indicates truncated output.

```
# symrdf list
```

```
Symmetrix ID: 000184500787
```

Local Device View														
		STATUS				M O D E S					RDF		S T A T E S	
Sym	RDF								R1 Inv	R2 Inv				
Dev	RDev	Typ:G	SA	RA	LNK	Mode	Dom	ACp	Tracks	Tracks	Dev	RDev	Pair	
01F	021	R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
020	022	R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
021	023	R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
022	024	R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
Total									-----	-----				
Track(s)									0	0				
MB(s)									0.0	0.0				

The following `symrdf list` command shows the Hop 1 view (sid 549) of the same Hop 1 and source Symmetrix devices. It also shows the R1 BCVs on Hop 1 that the example uses.

```
# symrdf list -sid 549
```

```
Symmetrix ID: 000184501549
```

DynamicRA ID: 00014501545

Local Device View														

		STATUS				M O D E S					RDF		S T A T E S	
Sym	RDF	-----				-----			R1 Inv	R2 Inv	-----		-----	
Dev	RDev	Typ:G	SA	RA	LNK	Mode	Dom	ACp	Tracks	Tracks	Dev	RDev	Pair	

021	01F	R2:1	RW	WD	RW	SYN	DIS	OFF	0	0	WD	RW	Synchronized	
022	020	R2:1	RW	WD	RW	SYN	DIS	OFF	0	0	WD	RW	Synchronized	
023	021	R2:1	RW	WD	RW	SYN	DIS	OFF	0	0	WD	RW	Synchronized	
024	022	R2:1	RW	WD	RW	SYN	DIS	OFF	0	0	WD	RW	Synchronized	

089	087	B-R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	

090	08E	B-R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
091	08F	B-R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	
092	090	B-R1:1	RW	RW	RW	SYN	DIS	OFF	0	0	RW	WD	Synchronized	

Creating a device group and adding devices to it are prerequisites for performing Symmetrix Automated Replication. The `symdg create` command creates a device group (Rdf1Grp). The `symld addall` command adds the SRDF standard devices from local Symmetrix 000184500787 to the group, using the `-range` option to limit the selections to those devices between 01F and 022.

```
# symdg create Rdf1Grp -type rdf1
# symld -g Rdf1Grp -sid 787 addall dev -range 01F:022
```

When EMC installs an SRDF configuration, the installers usually establish SRDF pairs at that time. The `symrdf query` command displays 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 any target-side host at this time. The link is operating in synchronous mode (indicated by SYN in the Mode column).

```
# symrdf query -g Rdf1Grp
```

```
Device Group (DG) Name: Rdf1Grp
DG's Type           : RDF1
DG's Symmetrix ID   : 000184500787
```

Source (R1) View					Target (R2) View					M O D E S			
-----					-----					-----			
Standard Logical Device	ST				LI	ST				M			
	A				N	A				O			
	T R1 Inv R2 Inv				K	T R1 Inv R2 Inv				d	RDF Pair		
	Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	e	Dom	ACp	STATE
-----					-----					-----			
DEV001	01F	RW	0	0	RW	021	WD	0	0	SYN	DIS	OFF	Synchronized
DEV002	020	RW	0	0	RW	022	WD	0	0	SYN	DIS	OFF	Synchronized
DEV003	021	RW	0	0	RW	023	WD	0	0	SYN	DIS	OFF	Synchronized
DEV004	022	RW	0	0	RW	024	WD	0	0	SYN	DIS	OFF	Synchronized
Total													
Track(s)		-----		0	-----		0	-----		0	-----		
MB(s)		0.0		0.0	0.0		0.0	0.0					

The `symdev list` command with the `-r1 -bcv` options displays those R1 BCV devices in the Hop 1 Symmetrix system (sid 549) that are not already part of a device group (N/AsstId) and which are free to be added to device group Rdf1Grp.

```
# symdev list -sid 549 -r1 -bcv
```

```
Symmetrix ID: 000184501549
```

Device Name		Directors		Device			
Sym	Physical	SA	:P DA	:IT Config	Attribute	Sts	Cap (MB)
087	Not Visible	04A:0	02A:C2	RDF1-BCV	N/Asst'd	RW	2063
088	Not Visible	04A:0	16B:D3	RDF1-BCV	N/Asst'd	RW	2063
089	Not Visible	04A:0	01A:C2	RDF1-BCV	N/Asst'd	RW	2063
08A	Not Visible	04A:0	01A:C4	RDF1-BCV	N/Asst'd	RW	2063
08B	Not Visible	04A:0	16B:D1	RDF1-BCV	N/Asst'd	RW	2063
08C	Not Visible	04A:0	02A:C4	RDF1-BCV	N/Asst'd	RW	2063
08D	Not Visible	04A:1	15B:D1	RDF1-BCV	N/Asst'd	RW	2063
08E	Not Visible	04A:1	15A:C4	RDF1-BCV	N/Asst'd	RW	2063
08F	Not Visible	04A:1	02B:D1	RDF1-BCV	N/Asst'd	RW	2063
090	Not Visible	04A:1	16A:C4	RDF1-BCV	N/Asst'd	RW	2063
091	Not Visible	04A:1	01B:D1	RDF1-BCV	N/Asst'd	RW	2063
092	Not Visible	04A:1	01B:C4	RDF1-BCV	N/Asst'd	RW	2063
093	Not Visible	04B:0	16A:D1	RDF1-BCV	N/Asst'd	RW	2063
094	Not Visible	04B:0	02B:C4	RDF1-BCV	N/Asst'd	RW	2063

The `symbcv` commands associate four R1 BCV devices (089, 090, 091, 092) with device group Rdf1Grp. The `-rdf` option indicates that these are remote BCVs (residing on Hop 1) and not local BCVs. SYMCLI assigns the devices the default logical names RBCV001, RBCV002, RBCV003, and RBCV004. Device 089 cannot be included with the `-range` option because the example does not use devices between 089 and 090.

```
# symbcv -g Rdf1Grp associate dev 089 -rdf
```

```
# symbcv -g Rdf1Grp associateall dev -range 090:092 -rdf
```

The `symmir establish` command initiates the creation of remote BCV pairs on Hop 1. Using the `-rdf` option targets the remote BCVs in the device group. The `-exact` option matches standard devices with BCV devices in the exact order that they were added to the device group. Thus, standard device 021 will be established with BCV 089, standard device 022 with BCV 090, and so forth.

```
# symmir -g Rdf1Grp establish -full -exact -rdf -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 remote BCV pairs on Hop 1 and their current state (SyncInProg).

```
# symmir query -g Rdf1Grp -rdf
```

```
Device Group (DG) Name :Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000184500787
Remote Symmetrix ID    : 000184501549
```

S T A N D A R D			S Y M M E T R I X			
Standard Device			BCV Device			State
Logical	Inv. Sym Tracks	Logical	Sym	Inv. Tracks	STD <=> BCV	
DEV001	021	0 RBCV001	089 *	59198	SyncInProg	
DEV002	022	0 RBCV002	090 *	59630	SyncInProg	
DEV003	023	0 RBCV003	091 *	56819	SyncInProg	
DEV004	024	0 RBCV004	092 *	57767	SyncInProg	
Total	-----			-----		
Track(s)	0			233414		
MB(s)	0.0			7294.2		

The `symmir verify` command with the `-rdf` option checks the state of the Hop 1 BCV pairs every 30 seconds until those BCV pairs are synchronized. Then the verify loop ends.

```
# symmir -g Rdf1Grp verify -rdf -i 30
```

None of the devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.

Not all devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.

All devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.

This subsequent `symmir query` command displays that the remote BCV pairs are now finished synchronizing (state is Synchronized). I/O to the source Symmetrix is copied from the source R1 standard devices to their remote R2 mirrors on the Hop 1 Symmetrix, and from those R2 mirrors to the BCVs.

```
# symmir query -g Rdf1Grp -rdf
```

```
Device Group (DG) Name : Rdf1Grp
DG's Type               : RDF1
DG's Symmetrix ID       : 000184500787
Remote Symmetrix ID     : 000184501549
```

```

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

Standard Device          BCV Device          State
-----
Logical      Inv.      Logical      Inv.      STD <=> BCV
              Sym Tracks              Sym  Tracks
-----
DEV001      021      0 RBCV001      089 *      0 Synchronized
DEV002      022      0 RBCV002      090 *      0 Synchronized
DEV003      023      0 RBCV003      091 *      0 Synchronized
DEV004      024      0 RBCV004      092 *      0 Synchronized

```

The `symrdf query` command displays the SRDF pair state of the Hop 1 BCVs with their R2 mirrors on Hop 2. Using the `-rbcv` option targets the Hop 1/Hop 2 SRDF pair. Establishing the BCV pairs had the effect of suspending any mirroring activity between the Hop 1 BCVs and their Hop 2 mirrors. Their state is Suspended.

```
# symrdf query -g Rdf1Grp -rbcv
```

```
Device Group (DG) Name : Rdf1Grp
DG's Type               : RDF1
DG's Symmetrix ID       : 000184500787
RBCV's Symmetrix ID     : 000184501549
```

```

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

Source (R1) View          Target (R2) View          M O D E S
-----
BCV      ST      LI      ST      M
Logical  A      N      A      o
Device  Dev  E  Tracks  Tracks  S  Dev  E  Tracks  Tracks  e  Dom ACp  RDF Pair
-----
RBCV001 089 NR      0      0 NR 087 WD      0      0 SYN DIS OFF Suspended
RBCV002 090 NR      0      0 NR 08E WD      0      0 SYN DIS OFF Suspended
RBCV003 091 NR      0      0 NR 08F WD      0      0 SYN DIS OFF Suspended
RBCV004 092 NR      0      0 NR 090 WD      0      0 SYN DIS OFF Suspended

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

```

The `symdg show` command displays detailed information about device group `Rdf1Grp` that is relevant to the local Symmetrix system (sid 787).

symdg show Rdf1Grp

Group Name: Rdf1Grp

```

Group Type           : RDF1
Valid                : Yes
Symmetrix ID         : 000184500787
Group Creation Time  : Mon Jun  4 16:15:41 2001
Vendor ID            : EMC Corp
Application ID       : SYMCLI

```

```

Number of STD Devices in Group : 4
Number of Associated GK's       : 0
Number of Locally-associated BCV's : 0
Number of Remotely-associated BCV's (STD RDF) : 4
Number of Remotely-associated BCV's (BCV RDF) : 0

```

Standard (STD) Devices (4):

{

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
DEV001	/dev/rdisk/clt0d0s2	01F		RW	2063
DEV002	/dev/rdisk/clt0d1s2	020		RW	2063
DEV003	/dev/rdisk/clt0d2s2	021		RW	2063
DEV004	/dev/rdisk/clt0d3s2	022		RW	2063

}

BCV Devices Remotely-associated (STD RDF) (4):

{

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
RBCV001	N/A	089		NR	2063
RBCV002	N/A	090		NR	2063
RBCV003	N/A	091		NR	2063
RBCV004	N/A	092		NR	2063

}

Device Group RDF Information

{

```

RDF Type           : R1
RDF (RA) Group Number : 1 (A)

Remote Symmetrix ID : 000184501549

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 : Direct Escon Fibre
RDF Link Domino      : Enabled
Prevent Automatic RDF Link Recovery : Enabled
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

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
}

```

The `symrdf query` command confirms the Synchronized state of the local SRDF pairs.

```
# symrdf query -g Rdf1Grp
```

```

Device Group (DG) Name: Rdf1Grp
DG's Type           : RDF1
DG's Symmetrix ID   : 000184500787

```

Source (R1) View					Target (R2) View					M O D E S		
Standard	ST				LI	ST				M		
Logical	A				N	A				o		
Device	T R1 Inv	R2 Inv	K		T R1 Inv	R2 Inv	d			RDF Pair		
Dev	E Tracks	Tracks	S	Dev	E Tracks	Tracks	e	Dom	ACp	STATE		
DEV001	01F RW	0	0 RW	021 WD	0	0	SYN	DIS	OFF	Synchronized		
DEV002	020 RW	0	0 RW	022 WD	0	0	SYN	DIS	OFF	Synchronized		
DEV003	021 RW	0	0 RW	023 WD	0	0	SYN	DIS	OFF	Synchronized		
DEV004	022 RW	0	0 RW	024 WD	0	0	SYN	DIS	OFF	Synchronized		

While part of a BCV pair, the R1 BCVs on Hop 1 are temporarily suspended from copying data to their remote R2 target mirrors. To resume mirroring between them, the example performs an instant split on the BCV pairs on Hop 1 and subsequently reestablishes these BCVs with their Hop 2 mirrors.

```
# symmir -g Rdf1Grp split -instant -rdf -noprompt
```

```

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

```

```
Remote 'Split' operation successfully executed for device group 'Rdf1Grp'.
```

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
# symmir -g Rdf1Grp verify -split -bg -i 10
```

None of the devices in group 'Rdf1Grp' have finished splitting in the background.

Not all devices in group 'Rdf1Grp' have finished splitting in the background.

All devices in group 'Rdf1Grp' have finished splitting in the background.

The `symrdf establish` command resumes the RDF links for the R1 BCV devices and initiates the propagation of data from the Hop 1 BCVs to their remote R2 target mirrors on Hop 2.

```
# symrdf -g Rdf1Grp establish -rbcv -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 verify` command with the `-rbcv` option displays a message every 30 seconds until all of the remote SRDF pairs (Hop 1 BCVs and their remote R2 mirrors) are synchronized.

```
# symrdf -g Rdf1Grp verify -rbcv -i 30
```

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.

The `symrdf query` command confirms that the SRDF pair state of the Hop 1 BCV devices and their Hop 2 mirrors is now Synchronized. The copy cycle from the source Symmetrix system to the R2 target mirrors on Hop 2 is now complete.

```
# symrdf -g Rdf1Grp query -rbcv
```

```
Device Group (DG) Name: Rdf1Grp
DG's Type           : RDF1
DG's Symmetrix ID    : 000184500787
RBCV's Symmetrix ID  : 000184501549
```

R E M O T E					S Y M M E T R I X							
Source (R1) View					Target (R2) View					M O D E S		
-----					-----					-----		
BCV	ST			LI	ST			M				
Logical	A			N	A			O				
Device	T R1 Inv	R2 Inv		K	T R1 Inv	R2 Inv		d				RDF Pair
Dev	E Tracks	Tracks		S Dev	E Tracks	Tracks		e	Dom	ACp		STATE
-----					-----					-----		
RBCV001	089 RW	0	0	RW 087 WD	0	0	0	SYN	DIS	OFF		Synchronized
RBCV002	090 RW	0	0	RW 08E WD	0	0	0	SYN	DIS	OFF		Synchronized
RBCV003	091 RW	0	0	RW 08F WD	0	0	0	SYN	DIS	OFF		Synchronized
RBCV004	092 RW	0	0	RW 090 WD	0	0	0	SYN	DIS	OFF		Synchronized
Total	-----		-----		-----		-----					
Track(s)	0		0		0		0					
MB(s)	0.0		0.0		0.0		0.0					

Initial setup for using Symmetrix Automated Replication also requires that the remote BCV pairs on Hop 1 be synchronized and that the remote SRDF pairs be suspended. Using the `symmir establish` command with the `-rdf` option re-establishes the Hop 1 BCV pairs and, at the same time, suspends the remote SRDF pairs (the Hop 1 BCVs with their R2 target devices on Hop 2).

```
# symmir -g Rdf1Grp establish -rdf -noprompt
```

```
'Incremental Establish' operation execution is in progress for device group
'Rdf1Grp'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device group
'Rdf1Grp'.
```

The `symmir verify` command with the `-rdf` option checks the state of the Hop 1 BCV pairs every 30 seconds until these remote BCV pairs are synchronized. When the verify loop ends, initial setup for Symmetrix Automated Replication is complete for the multihop, no-BCV configuration.

```
# symmir -g Rdf1Grp verify -rdf -i 30
```

```
None of the devices in group 'Rdf1Grp' are in the 'Synchronized or Restored'
state.
```

```
Not all devices in group 'Rdf1Grp' are in the 'Synchronized or Restored'
state.
```

```
All devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.
```

The requirements for beginning a replicate session in a multihop, no-BCV configuration have now been met:

- ◆ Local SRDF pairs are synchronized.
- ◆ BCV pairs on Hop 1 are synchronized.
- ◆ Remote SRDF pairs are suspended.

Refer to the previous examples for creating a replicate options file and starting a replicate session.

B.4 Example 4: Accessing Concurrent Non-SRDF/AR BCVs while Running SRDF/AR (Single-Hop Configuration)

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.

```
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.

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
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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.

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...
S1 cycle(s) complete; exiting symreplicate...
```

B.5 Example 5: Accessing Concurrent Non-SRDF/AR BCVs while Running SRDF/AR (Multihop Configuration)

The hardware setup for this multihop SRDF/AR configuration consists of two hosts—one connected to a source Symmetrix, and the other connected to a remote target Symmetrix (sid 33) at Hop 2. Some commands are issued from the source-side host (whose system prompt is src#), and some from the target-side host (tgt#). The SRDF/AR device group (sar) and the creation of the multihop 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 source-side host has a device *group* defined for running SRDF/AR, you also need to create a device *file* that allows the source-side host to manipulate the SRDF/AR BCV pairs located on the remote Symmetrix unit (sid 33) at Hop 2. The following `symmir query` command from the source-side host (src#) examines the status of the SRDF/AR BCV pairs on Hop 2 that were defined previously in device file `devfile`.¹

```
src# symmir -f devfile query -sid 33
```

```
Device File Name      : devfile
Device's Symmetrix ID : 000000005233
```

Standard Device			BCV Device			State	
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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	
MB(s)						0.0	

Legend:

(*): The paired BCV device is associated with this group.

1. The device file `devfile` defines the following SRDF/AR pairs:

```
0001 0031
0002 0032
0003 0033
0004 0034
0005 0035
```

On the target-side host (tgt#), create a device group to manipulate the non-SRDF/AR BCVs. The `syndg create` command creates an R2 type device group (mbcv). The `syml d 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).

```
tgt# syndg create mbcv -type rdf2
tgt# syml d -g mbcv addall dev -range 0001:0005 -sid 33
tgt# symbcv -g mbcv associateall dev -range 0043:0047
```

The `symmir establish` command fully establishes the BCV pairs in the exact order that they were defined in the device group.

```
tgt# 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 displays the status of the non-SRDF/AR BCV pairs (SyncInProgress).

```
tgt# symmir -g mbcv query
```

```
Device Group (DG) Name: mbcv
DG's Type           : RDF2
DG's Symmetrix ID    : 000000005233
```

Standard Device			BCV Device			State	
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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 `symmir query` command 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).

```
tgt# symmir -g mbcv query -multi
```

```
Device Group (DG) Name: mbcv
DG's Type           : RDF2
DG's Symmetrix ID    : 000000005233
```

Standard Device			BCV Device			State
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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 checks the state of the BCV pairs in the device group every five seconds until the non-SRDF/AR BCV pairs are synchronized.

```
tgt# 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 splits the non-SRDF/AR BCV pairs.

```
tgt# 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 example needs to establish and split the SRDF/AR BCV pairs now so that they are the last pairs to be manipulated before beginning the `symreplicate` cycle. This manipulation is only required prior to starting SRDF/AR. The following `symmir establish` command from the source-side host establishes the SRDF/AR BCV pairs in device file devfile.

```
src# 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 splits the SRDF/AR BCV pairs in device file `devfile`.

```
src# 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, using the configuration of devices defined in the device group `sar` and multihop copy options defined in a file called `rep.txt` (for file content, refer to the `rep_opt.txt` file defined in Example 2).

```
src# 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 target-side 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.

```
tgt# 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 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.

```
tgt# 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 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.

```
tgt# symmir -g mbcv query -multi
```

```
Device Group (DG) Name: mbcv
DG's Type           : RDF2
DG's Symmetrix ID   : 000000005233
```

Standard Device			BCV Device			State
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. 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 to split the non-SRDF/AR BCV pairs without the `-skip` option fails.

```
tgt# 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 to split the non-SRDF/AR BCV pairs using the `-skip` option succeeds.

```
tgt# 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'.

B.6 Example 6: 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: 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...
```

