
Using SYMCLI to Perform SRDF/AR

Abstract

This white paper introduces the SRDF/AR add-on to the SRDF product family of base solutions. SRDF/AR allows you to automate data copies across RDF links in an open systems environment.

Published 1/25/2005

Copyright © 2005 EMC Corporation. All rights reserved.

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.

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

Part Number 300-000-078 REV K

Table of Contents

Introduction	5
Purpose and Scope	5
Related Documentation	5
Practical Uses	5
Using SRDF/AR	6
SRDF/AR Single-Hop Configuration	6
SRDF/AR Multi-Hop Configuration	7
Multi-Hop Configuration without the Hop 2 BCV	7
Multi-Hop Configuration with the Hop 2 BCV	7
Initial Setup Tasks for Using SRDF/AR	8
Building a Device Group for a Replicate Session	8
Defining Copy Cycle Parameters for a Replicate Session	9
Setting Up Pairings and Pair States for a Replicate Session	11
Starting and Stopping a Replicate Session	12
Restarting a Replicate Session in a Clustered SRDF/AR Environment	12
Error Handling for a Replicate Session	13
Using Protected BCV Establish When a BCV Has Two Mirrors	15
Setting TimeFinder Clone Emulation	15
Accessing a Concurrent BCV While Running SRDF/AR	15
Using a Composite Group for Applications Spanning Multiple Symmetrix Arrays (SRDF/AR Single-Hop)	17
Using a Composite Group for Applications Spanning Multiple Symmetrix Arrays (SRDF/AR Multi-Hop)	19
Example 1: SRDF/AR Single-Hop Configuration	21
Example 2: SRDF/AR Multi-Hop Configuration with BCVs at Hop 2	33
Example 3: Setting Up an SRDF/AR Single-Hop Configuration Using a Composite Group	35
Example 4: Setting Up an SRDF/AR Multi-Hop Configuration Using a Composite Group	43
Example 5: Accessing Concurrent Non-SRDF/AR BCVs While Running SRDF/AR (Single-Hop Configuration)	48

**Example 6: Accessing Concurrent Non-SRDF/AR BCVs While Running
SRDF/AR (Multi-Hop Configuration)..... 52**

Example 7: Restarting a Replicate Session When Devices Are Locked 58

Introduction

The SRDF® product family of base solutions includes an SRDF/AR add-on that allows you to automate data replication across RDF links in an open systems environment. SRDF/AR allows you to set up remote mirroring on an automatic basis according to your own predefined copy cycle (every hour, for example).

TimeFinder®/Mirror software (referred to as “TimeFinder in this document) works with the SRDF product family (referred to as “SRDF” in this document) to mirror data from a source Symmetrix® system to a remotely-associated BCV located across the hall or across the globe, creating 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/AR multi-hop configuration* (multi-hop mirroring to a third site) provides a recovery solution for component or site failures between remotely-mirrored devices. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster.

SRDF Adaptive Copy replication allows you to mirror to a Symmetrix system in a third location only the data that has changed since the last update. 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. Multi-hop 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.

Purpose and Scope

This document describes SRDF/AR functionality in versions of EMC® Solutions Enabler up through version 6.0 and Engenuity™ version 5x71. 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*.

Related Documentation

The following white papers provide supporting information for the concepts discussed in this paper:

- *Using SYMCLI to Perform Control Operations with SRDF Family Products* (P/N 300-000-076)
- *Using SYMCLI to Perform TimeFinder/Mirror Control Operations* (P/N 300-000-074)

Practical Uses

SRDF/AR allows you to set up remote mirroring on an automatic basis according to your own predefined copy cycle frequency and duration. 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 multi-hop 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 SRDF/AR provide 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.

Using SRDF/AR

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.

This paper describes how to use SRDF/AR to copy data in a single- or multi-hop configuration. For information about copying in multi-hop environments *without* SRDF/AR, refer to the white paper *Using SYMCLI to Perform Control Operations with SRDF Family Products* (P/N 300-000-076).

SRDF/AR Single-Hop Configuration

Figure 1 shows that 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.

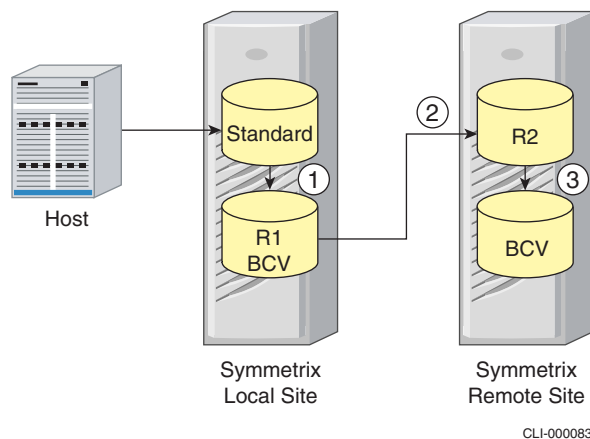


Figure 1. Automated Data Copying in a Single-Hop Configuration

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.

SRDF/AR Multi-Hop Configuration

Figure 2 shows that the copy path for a multi-hop 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).

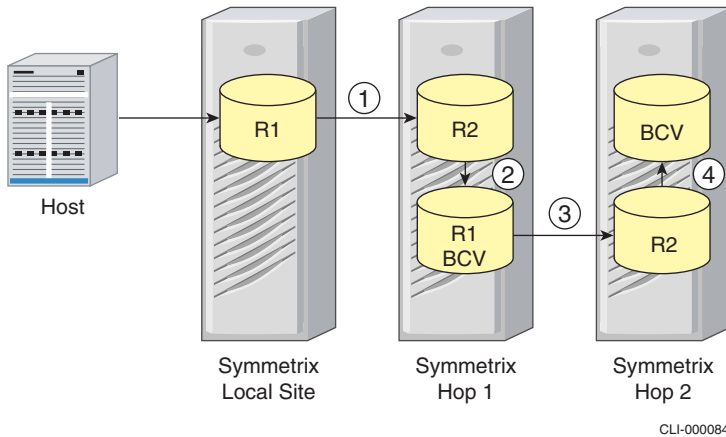


Figure 2. Automated Data Copying in a Multi-Hop Configuration

The multi-hop 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.

Multi-Hop Configuration without the Hop 2 BCV

Automated data replication in a multi-hop 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.

Multi-Hop 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.

Initial Setup Tasks for Using SRDF/AR

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

1. Build a device group (or composite group¹) that includes devices for the configuration.
2. Create a replicate options file that specifies your configuration (single-hop or multi-hop) and defines the copy cycle parameters for your replicate session.
3. Determine how you want pairs matched during setup: default, exact, or optimized.
4. Initiate the replicate session using the `symreplicate` command and specifying the name of your replicate options file and the pairing setup that you want.

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 *multi-hop 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. 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
```

¹ Beginning with Solutions Enabler version 5.4, you can use composite group support to implement an SRDF/AR single- or multi-hop configuration in which your application spans multiple hosts and multiple Symmetrix arrays. Refer to the sections in this paper that describe using composite groups for this type of application.

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 multi-hop configuration, respectively.

The *TrueOrFalse* value specifies whether the Hop 2 Symmetrix has BCVs that participate in the multi-hop 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.

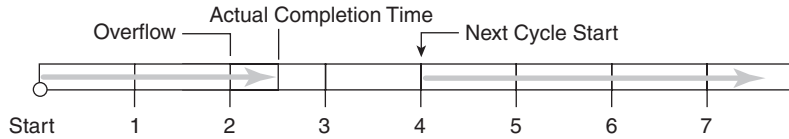
The *DelayTime* value specifies the minimum amount of time to wait between the end of one cycle and the beginning of the next cycle. Even if a cycle runs longer than the specified cycle time and the overflow behavior is set to `immediate`, the system waits for the duration of the *DelayTime* value before 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, it is recommended that you start 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:

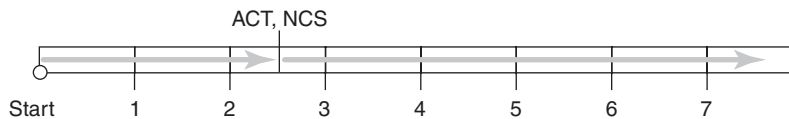
- *CycleTime=60, DelayTime=0, OverflowBehavior=NEXT*
New cycle every hour, if possible; otherwise, every two or three hours based on data throughput and size.
- *CycleTime=0, DelayTime=60*
Complete the first copy cycle, then wait an hour; then another cycle and a one-hour wait; and so forth.

Figure 3 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.

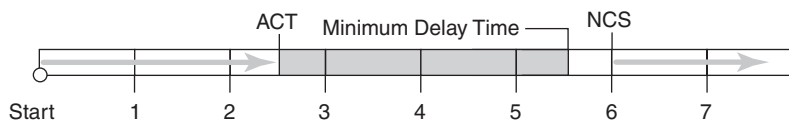
1. Cycle Time = 2, Delay Time = 0, Overflow = Next



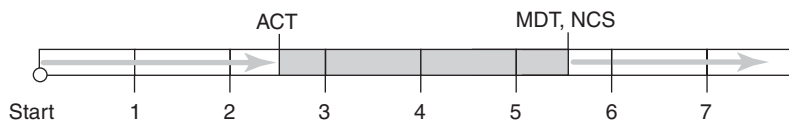
2. Cycle Time = 2, Delay Time = 0, Overflow = Immediate



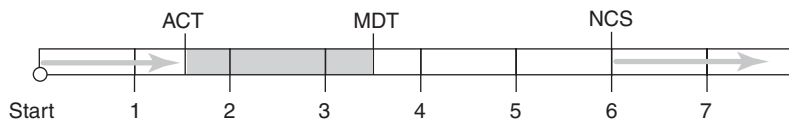
3. Cycle Time = 2, Delay Time = 3, Overflow = Next



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

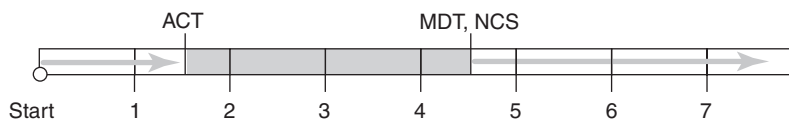


Figure 3. Copy Cycle Results for Different Replicate Cycle Parameter Settings

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. At the end of a cycle, the system waits three minutes and then begins another copy cycle, regardless of the overflow setting.

Setting Up Pairings and Pair States for a Replicate Session

SRDF/AR requires that SRDF and BCV pairs be in a certain state prior to starting 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 begin a replicate session in the *multi-hop 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.

Prior to Solutions Enabler version 5.4, you had to perform these setup steps manually through a sequence of SRDF and TimeFinder commands. Beginning with version 5.4, you can put the pairs in these states automatically by using the `symreplicate setup` command or the `-setup` option with the `symreplicate start` command.

The `setup` command sets up the required pair states, executes *one* cycle, which may take some time, and then exits. The `symreplicate` options file tells the setup utility which hop type (single or multi) to set up; this same options file can be used for both the `setup` command and a subsequent `start` command. The `setup` command executes just one cycle, regardless of the number of cycles specified for the `symreplicate` session in the options file.

```
symreplicate -g MyDevGroup setup -options MyOpFile
```

The `start` command with the `-setup` option sets up the required pair states, and if successful, *begins* the `symreplicate` session. The `symreplicate` options file defines the hop type and the copy cycle parameters that you have chosen for the session.

```
symreplicate -g MyDevGroup start -options MyOpFile -setup
```

The default setup operation provides no optimization in the device pairings. SRDF/AR simply places the devices in the required initial state while preserving as many pairing relationships as possible by not splitting and establishing any devices that are already in the initial state. If the standard devices have no history with the BCVs in the device group or composite group, SRDF/AR pairs them according to the first available BCV that has no history with any of the compatible standard devices in the group.

Including the `-optimize` option with either of the commands creates device pairings that are optimized within a Symmetrix array without regard for the RDF (RA) groups that the devices belong to. SRDF/AR splits any BCV pairs that are currently established and pairs the group's entire set of standard devices and BCVs using optimization rules. Including the `-optimize_rag` option creates device pairings that are optimized within RDF (RA) groups on a Symmetrix array.

Including the `-exact` option provides pairing such that the standard devices and BCVs are paired in the exact order that they were added to the device group or composite group (that is, the first standard device with the first BCV device), without any optimization.

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 EMC Solutions Enabler version 5.2, by recovering the existing device locks using the `-recover` option (see Example 7).

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
```

The following `symreplicate start` command sets up device pairing and pair states and initiates the first replicate copy cycle using the parameters defined in the replicate options file. The `-consistent` option ensures a consistent split of the appropriate BCV pairs. Subsequent commands query the replicate session, stop it, and restart it.

```
symreplicate -g RegularGrp start -options opfile.txt -setup -consistent
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 *EMC Solutions Enabler Symmetrix SRDF CLI Product Guide*.

Restarting a Replicate Session in a Clustered SRDF/AR Environment

Beginning with EMC 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 array 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 array 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. By default, SRDF/AR uses the device group or composite group name as the name of the log file. Optionally, you can also specify the `-log` option and a file name (such as `log2`). For example:

```
symreplicate -g symrep start -sid 202 -log log2
symreplicate -cg 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` or `-cg CompositeGroupName`) 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 array. 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
SYMREP	Device group
NEW	Composite group

The `symreplicate delete` command with the `-sid` option and the `-log` option, `-g` option, or `-cg` option allows you to delete a log file from the SFS on a Symmetrix array. 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.

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.²

```
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
```

² The `TIME_LIMIT` and `SLEEP_TIME` options in this list should not be set unless directed by an EMC representative.

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 *EMC 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), the `steperror` script file will not execute.

Releasing Device External Locks If a Replicate Session Terminates Abnormally

To prevent other applications from altering device states while a replicate session is active, device external locks are held during the session. If the session terminates abnormally (for example, the system crashes or a component fails), you may need to release the device locks manually. Beginning with EMC Solutions Enabler version 6.0 and Enginuity version 5x71, you can do so using the `symreplicate release` command.

```
symreplicate release -g symrep
```

Any device external locks associated with devices in the device group or composite group that were locked by a previous replicate session are released. The `-force` option is required if the release operation is requested in a clustered environment on a host that did not initiate the target replicate session and the status of this session cannot be determined, or if the targeted set of devices to be unlocked was not locked by the previous replicate session (in which case, the lock-holder ID does not match the previous session's lock-holder ID) and you want to release the device locks with the previous session's lock-holder ID.

Using Protected BCV Establish When a BCV Has Two Mirrors

You can include an entry in your replicate options file that sets up protected BCV establish operation for a single-hop or a multi-hop configuration.

```
SYMCLI_REPLICATE_PROTECT_BCVS=Protection
```

The protection value that you choose indicates how symreplicate should perform TimeFinder establish operations when two-way mirrored BCV devices are used. *Protection* can be one of the following values:

- NONE (the default). The value NONE performs a normal TimeFinder establish operation.
- LOCAL. Performs protected BCV establish for the local BCV pairs only in a single-hop configuration.
- REMOTE. Performs protected BCV establish for the remote BCV pairs only in a single-hop configuration.
- FIRST_HOP. Performs protected BCV establish for the Hop-1 BCV pairs only in a multi-hop configuration.
- SECOND_HOP. Performs protected BCV establish for the Hop-2 BCV pairs only in a multi-hop configuration.
- BOTH. Performs protected BCV establish for both local and remote BCV pairs in a single-hop configuration. Does the same for both Hop-1 and Hop-2 BCV pairs in a multi-hop configuration.

Setting TimeFinder Clone Emulation

Beginning with Solutions Enabler version 6.0, TimeFinder automatically maps TimeFinder/Mirror commands to existing TimeFinder/Clone commands when the BCV is a RAID-5 BCV, provided that no non-RAID-5 BCVs exist in the devices being controlled. In a mixed environment, you must explicitly set clone emulation mode. SRDF/AR allows you to include an entry in your replicate options file to indicate that Time Finder clone emulation is enabled (value is TRUE). The default value is FALSE.

```
SYMCLI_REPLICATE_TF_CLONE_EMULATION=TRUE
```

For more information about TimeFinder clone emulation, refer to the white paper *Using SYMCLI to Perform TimeFinder/Mirror Control Operations* (P/N 300-000-074).

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 host³ 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 a 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 a SRDF/AR BCV during the SRDF/AR copy cycle.

³ A device can be included in only one device group on the same host.

In Figure 4, BCV1 and BCV2 are concurrent BCVs, as are BCV3 and BCV4. But BCV2 and BCV4 are not in the device group participating in the SRDF/AR copy cycle. By defining a device file on Host 1 that includes the standard device and BCV1, you can establish this BCV pair during the copy cycle, split the pair, and then access the data on BCV2 without interfering with the copy cycle.

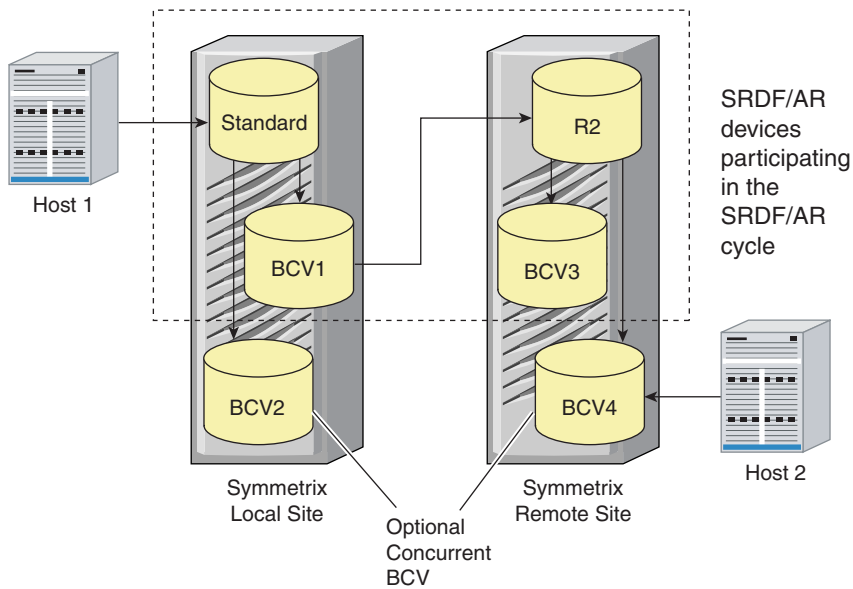


Figure 4. Accessing Concurrent BCVs in a Single-Hop Configuration

Similarly, you can control and access BCV4 during the SRDF/AR copy cycle by defining a separate device group (or device file) on Host 2 that includes the R2 and BCV4.

Figure 5 illustrates a multi-hop configuration with concurrent BCVs that are not in the device group that participates in the SRDF/AR copy cycle (refer to Example 6).

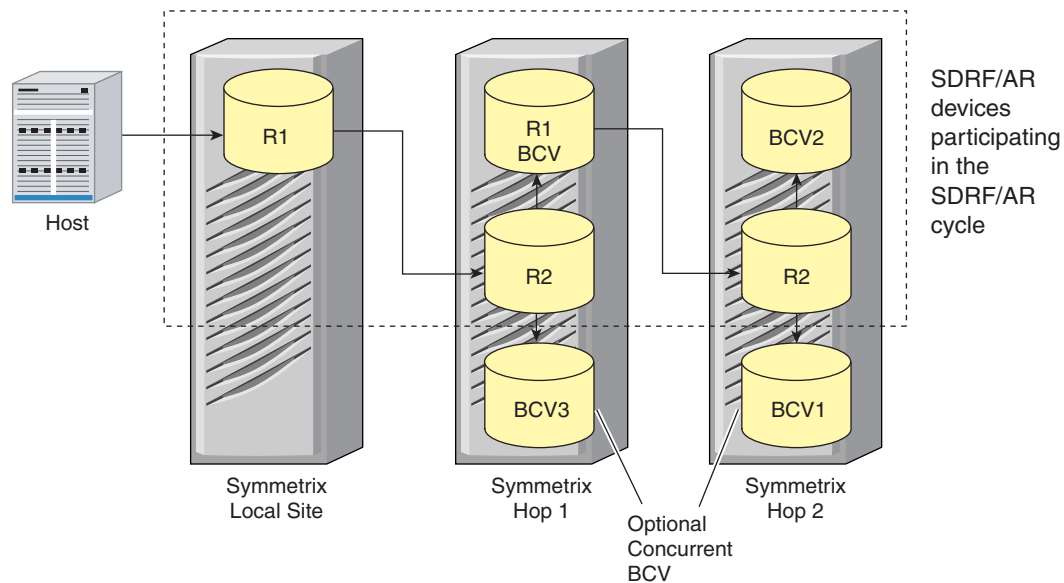


Figure 5. Accessing a Concurrent BCV in a Multi-Hop Configuration

Using a Composite Group for Applications Spanning Multiple Symmetrix Arrays (SRDF/AR Single-Hop)

A *composite group* is a user-defined group of devices that can span multiple Symmetrix arrays. Recall that a device group of the type RDF1 or RDF2 is limited to a single Symmetrix array. Beginning with Solutions Enabler version 5.4, SRDF/AR incorporates support for composite groups. For example, you can use a composite group and distributed consistent split to create a DBMS-restartable copy of a database that spans multiple hosts and multiple Symmetrix arrays.

Figure 6 illustrates a single-hop configuration in which the source devices span two hosts and two Symmetrix arrays (A and C). A composite group is defined on whichever host is the controlling host⁴ and includes standard devices and R1 BCV devices from local Symmetrix arrays A and C, as well as the BCVs from remote Symmetrix arrays B and D.

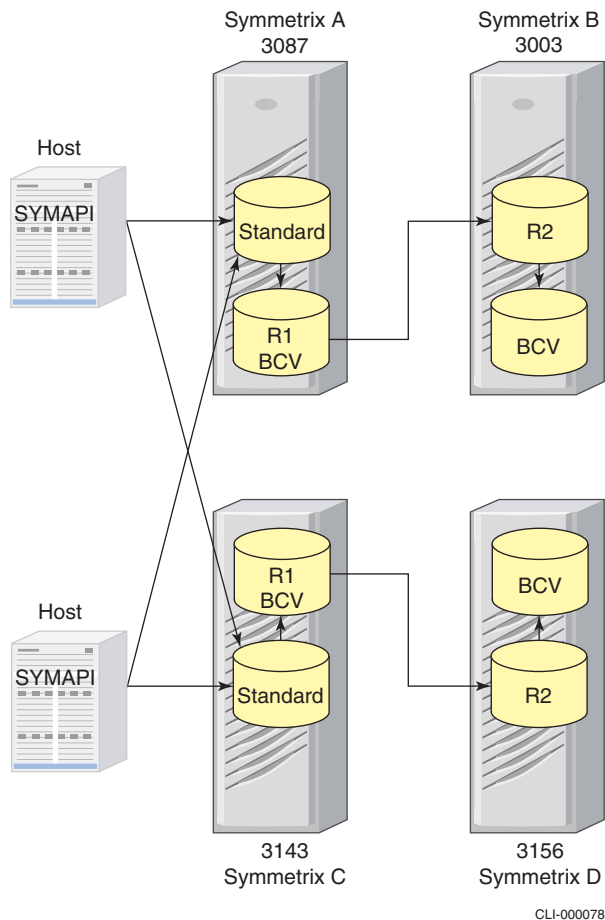


Figure 6. Controlling Source Devices that Span Multiple Symmetrix Arrays

When established as BCV pairs, the local R1 BCVs are suspended from copying data to their remote (R2) targets. Consequently, the SRDF/AR control software must split the local BCV pairs before it can establish

⁴ If Group Name Services (GNS) is enabled and the GNS daemon is running on the local hosts, the composite group definition is propagated to the Symmetrix arrays and to other hosts connected to these arrays. For details, refer to the white paper *Using SYMCLI and GNS to Propagate Group Definitions to Multiple Hosts* (P/N 300-001-384).

the SRDF pairs (R1 BCVs with their respective R2 devices). Subsequently, SRDF/AR copies data from the R2 devices to the remote BCVs.

The following steps outline the setup required for performing a replicate session in the single-hop configuration shown in Figure 6. Note that when adding devices to a composite group, you always specify the Symmetrix IDs (`-sid`) of the local Symmetrix arrays, regardless of whether the devices being added are local or remote.

1. From the controlling host, create a Regular type composite group (for example, `MyGrp`). The composite group can be any type when using the single-hop configuration.

```
symcgr create MyGrp -type regular
```
2. Add to the composite group the standard devices on Symmetrix A (3087) and Symmetrix C (3143).

```
symcgr -cg MyGrp -sid 3087 add dev 27C
symcgr -cg MyGrp -sid 3143 add dev 004
```
3. Associate the local R1 BCV devices on Symmetrix arrays A and C with the composite group.

```
symbcv -cg MyGrp -sid 3087 associate dev 2E4
symbcv -cg MyGrp -sid 3143 associate dev 048
```
4. Associate the remote BCV devices on Symmetrix arrays B and D. If there is more than one RDF group on a local Symmetrix array (in this case, 3087 and 3143), you must include the RDF group number of the local source devices (the group number of R1 BCVs 2E4 and 048). When adding multiple devices on a single Symmetrix array, specifying the RDF group number creates the RDF link for each R1/R2 pair.

```
symbcv -cg MyGrp -sid 3087 associate dev 2E9 -rdf -bcv -rdfig 2
symbcv -cg MyGrp -sid 3143 associate dev 045 -rdf -bcv -rdfig 1
```
5. Create a replicate options file (for example, `OpFile.txt`) that defines the hop type and your copy cycle parameters.

```
vi OpFile.txt
```

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=60
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
```
6. Run the setup utility to set up the device pairings and pair states necessary to start the replicate session. Use the `-optimize` option to optimize device pairings within each Symmetrix array without regard for the RDF (RA) group that a device belongs to, or the `-optimize_rag` option if you want to optimize device pairings within RDF (RA) groups of each Symmetrix array, or the `-exact` option for exact device pairings. Specify your replicate options file so that setup will know the hop type. The following command places the device pairs in the correct state for the replicate session.

```
symreplicate -cg MyGrp setup -options OpFile.txt -optimize
```
7. Start a replicate session. Specify your replicate options file so that the replicate session knows your copy cycle parameters.

```
symreplicate -cg MyGrp start -options OpFile.txt -consistent
```

The `-consistent` option is used to achieve a distributed consistent split during the copy cycle for all BCV pairs on the local Symmetrix arrays (3087 and 3143).

Using a Composite Group for Applications Spanning Multiple Symmetrix Arrays (SRDF/AR Multi-Hop)

As described in the previous section, you can use composite group support beginning with Solutions Enabler version 5.4 to implement a multi-hop configuration in which an application such as a database spans multiple hosts and multiple Symmetrix arrays. Figure 7 illustrates a multi-hop configuration in which the source devices span two hosts and two Symmetrix arrays (A and C). A composite group is defined on the controlling host⁵ and includes R1 devices from local Symmetrix arrays A and C, R1 BCV devices from Symmetrix arrays B and D, and BCV devices from Symmetrix arrays E and F.

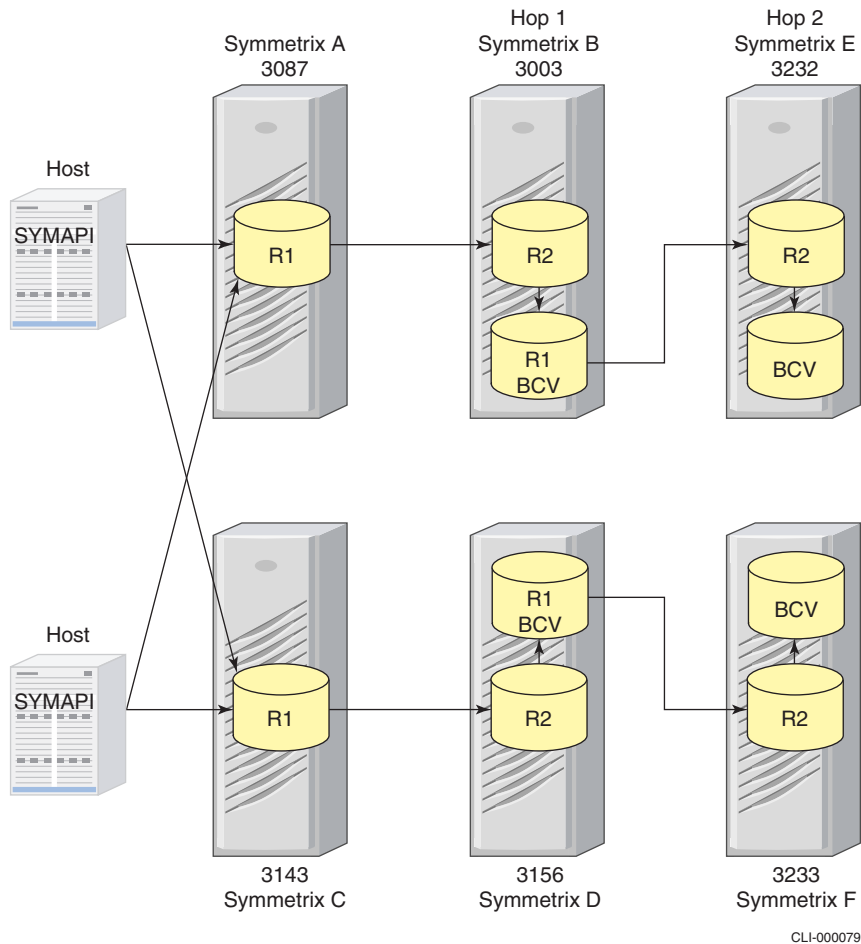


Figure 7. Multi-Hop Configuration in a Distributed Database Environment

When established as BCV pairs, the remote R1 BCVs at Hop 1 are suspended from copying data to their R2 targets at Hop 2. Consequently, SRDF/AR must split the Hop 1 BCV pairs before it can establish the SRDF pairs (R1 BCVs with their respective R2 devices at Hop 2). Subsequently, SRDF/AR copies data from those R2 devices at Hop 2 to the BCVs there.

The following steps outline the setup required for performing a replicate session in the distributed database environment shown in Figure 7:

⁵ If Group Name Services (GNS) is enabled and the GNS daemon is running, the composite group definition is propagated to the Symmetrix arrays and to other hosts connected to those arrays.

1. From the controlling host, create an RDF1 type composite group (for example, MyGrp).

```
symcgr create MyGrp -type rdf1
```
2. Add to the composite group those standard devices on Symmetrix A (3087) and Symmetrix C (3143) that hold the database.

```
symcgr -cg MyGrp -sid 3087 add dev 280
symcgr -cg MyGrp -sid 3143 add dev 004
```
3. Remotely associate the R1 BCV devices at Hop 1 (remote Symmetrix arrays B and D) using the `-rdf` option. If there is more than one RDF group on a local Symmetrix array (in this case, 3087 and 3143), you must include the RDF group number of the local source devices (R1 devices 280 and 004). When adding multiple devices on a single Symmetrix array, specifying the RDF group number creates the RDF link for each R1/R2 pair.

```
symbcv -cg MyGrp -sid 3087 associate dev 2E4 -rdf -rdfg 2
symbcv -cg MyGrp -sid 3143 associate dev 2E9 -rdf -rdfg 1
```
4. Remotely associate the BCV devices at Hop 2 (remote Symmetrix arrays E and F) using the `-rrdf` option. To define the path of the RDF link, you must always include the group number of the local source devices as specified with the `-rdfg` option.⁶

```
symbcv -cg MyGrp -sid 3087 associate dev 2E0 -rrdf -rdfg 2
symbcv -cg MyGrp -sid 3143 associate dev 2E5 -rrdf -rdfg 1
```
5. Create a replicate options file (for example, OpFile.txt) that defines the hop type and your copy cycle parameters.

```
vi OpFile.txt

SYMCLI_REPLICATE_HOP_TYPE=MULTI
SYMCLI_REPLICATE_CYCLE=60
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
```
6. Run the setup utility to set up the device pairings and pair states necessary to start the replicate session. Use the `-optimize` option to optimize device pairings within each Symmetrix array without regard for the RDF (RA) group that a device belongs to, or the `-optimize_rag` option if you want to optimize device pairings within RDF (RA) groups of each Symmetrix array, or the `-exact` option for exact device pairings. Specify your replicate options file so that setup will know the hop type. The following command places the device pairs in the correct state for the replicate session.

```
symreplicate -cg MyGrp setup -options OpFile.txt -optimize
```
7. Start a replicate session. Specify your replicate options file so that the replicate session knows your copy cycle parameters.

```
symreplicate -cg MyGrp start -options OpFile.txt -consistent
```

The `-consistent` option is used to achieve a distributed consistent split during the copy cycle for all BCV pairs on the Hop1 Symmetrix arrays.

⁶ If you add the Hop 2 BCVs to the composite group before adding the Hop 1 BCVs, include the `-remote_rdfg` option also.

Example 1: SRDF/AR Single-Hop Configuration

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a HP-UX host connected to a source Symmetrix (sid 161). All commands are issued from the source-side host. The example uses the following Symmetrix systems to create the single-hop environment:

- Local Source Symmetrix (sid 35): standard devices 56-6E; R1 BCV devices 182-19A
- Remote Target Symmetrix (sid 41): BCV devices 137-14F

The `symcfg list` command displays the Symmetrix arrays attached to this host. Symmetrix arrays 000187900035 and 000187900041 are configured as a single-hop configuration.

```
# symcfg list
```

S Y M M E T R I X						
SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000000003143	Local	3630	5267	4096	209	447
000000005232	Local	8230	5568	2048	6	652
000184500160	Local	8430	5568	12288	132	2054
000187700079	Local	DMX2000P	5670	51200	189	3326
000187900035	Local	DMX800	5670	6144	204	812
000000003156	Remote	3630	5267	4096	0	448
000000005231	Remote	8230	5568	4096	0	714
000000005233	Remote	8230	5568	6144	0	122
000000006201	Remote	DMX2000P	5670	16384	0	824
000184502898	Remote	8530	5568	12288	0	1139
000187700067	Remote	2000P-M2	5670	51200	0	3762
000187900041	Remote	DMX800	5670	6144	0	721

The `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 those devices that are used in the example. The N/Grp'd attribute means that these devices are not already part of a device group and are free to be added to device group `symrep`.

```
# sympd list -sid 35
```

Symmetrix ID: 000187900035

Device Name	Directors			Device			
Physical	Sym	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)
/dev/rdisk/c15t2d0s2	0008	01C:1	01A:CC	2-Way Mir	N/Grp'd	RW	3
/dev/rdisk/c15t2d1s2	0009	01C:1	16B:CC	2-Way Mir	N/Grp'd	RW	3
/dev/rdisk/c15t2d2s2	0056	01C:1	01B:CD	2-Way Mir	N/Grp'd (M)	RW	12946
/dev/rdisk/c15t2d3s2	0059	01C:1	16A:C5	2-Way Mir	N/Grp'd (M)	RW	8631
/dev/rdisk/c15t2d4s2	005B	01C:1	16B:C2	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d5s2	005C	01C:1	01A:CA	2-Way Mir	N/Grp'd	RW	4315

/dev/rdisk/c15t2d6s2	005D	01C:1	16A:C7	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d7s2	005E	01C:1	01B:C7	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d8s2	005F	01C:1	16B:C4	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d9s2	0060	01C:1	01A:C6	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d10s2	0061	01C:1	16A:C9	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d11s2	0062	01C:1	01B:C9	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d12s2	0063	01C:1	16B:C6	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d13s2	0064	01C:1	01A:C8	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d14s2	0065	01C:1	02A:CB	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d15s2	0066	01C:1	15B:CB	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d16s2	0067	01C:1	02B:CA	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d17s2	0068	01C:1	15A:CC	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d18s2	0069	01C:1	02A:CD	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d19s2	006A	01C:1	15B:C1	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d20s2	006B	01C:1	02B:C0	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d21s2	006C	01C:1	15A:CE	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d22s2	006D	01C:1	02A:C1	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c15t2d23s2	006E	01C:1	15B:C3	2-Way Mir	N/Grp'd	RW	4315

The `symld` command adds one or more standard devices to the device group; the `-range` option can be used with the `addall` action to limit the selection to the devices that are within the specified range (for example, devices 56 through 6E).

```
# symld -g symrep addall dev -range 56:6E -sid 35
```

The `symdev list` command with the `-r1 -bcv` options displays those R1 BCV devices in the local Symmetrix system (sid 35) that are not already part of a device group (N/Asst'd) and which are free to be added to the device group. The display below has been edited to show those devices that are used in the example.

```
# symdev list -r1 -bcv -sid 35
```

Symmetrix ID: 000187900035

Device Name		Directors		Device				Cap
Sym	Physical	SA	:P DA :IT	Config	Attribute	Sts	(MB)	
0182	Not Visible	?????	16A:CB	RDF1-BCV+Mir	N/Asst'd (M)	NR	12946	
0183	Not Visible	?????	01B:CB	RDF1-BCV+Mir	N/Asst'd (m)	NR	-	
0184	Not Visible	?????	16B:CA	RDF1-BCV+Mir	N/Asst'd (m)	NR	-	
0185	Not Visible	?????	01A:CC	RDF1-BCV+Mir	N/Asst'd (M)	NR	8631	
0186	Not Visible	?????	16A:CD	RDF1-BCV+Mir	N/Asst'd (m)	NR	-	
0187	Not Visible	?????	01B:C1	RDF1-BCV+Mir	N/Asst'd	NR	4315	
0188	Not Visible	?????	16B:C0	RDF1-BCV+Mir	N/Asst'd	NR	4315	
0189	Not Visible	?????	01A:CE	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018A	Not Visible	?????	16A:C1	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018B	Not Visible	?????	01B:C3	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018C	Not Visible	?????	16B:C8	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018D	Not Visible	?????	01A:C0	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018E	Not Visible	?????	16A:C3	RDF1-BCV+Mir	N/Asst'd	NR	4315	
018F	Not Visible	?????	01B:CD	RDF1-BCV+Mir	N/Asst'd	NR	4315	
0190	Not Visible	?????	16B:CC	RDF1-BCV+Mir	N/Asst'd	NR	4315	
0191	Not Visible	?????	01A:C2	RDF1-BCV+Mir	N/Asst'd	NR	4315	
0192	Not Visible	?????	16A:C5	RDF1-BCV+Mir	N/Asst'd	NR	4315	

0193	Not Visible	???:?	01B:C5	RDF1-BCV+Mir	N/Asst'd	NR	4315
0194	Not Visible	???:?	16B:CE	RDF1-BCV+Mir	N/Asst'd	NR	4315
0195	Not Visible	???:?	01A:C4	RDF1-BCV+Mir	N/Asst'd	NR	4315
0196	Not Visible	???:?	16A:C7	RDF1-BCV+Mir	N/Asst'd	NR	4315
0197	Not Visible	???:?	01B:C7	RDF1-BCV+Mir	N/Asst'd	NR	4315
0198	Not Visible	???:?	16B:C2	RDF1-BCV+Mir	N/Asst'd	NR	4315
0199	Not Visible	???:?	01A:CA	RDF1-BCV+Mir	N/Asst'd	NR	4315
019A	Not Visible	???:?	16A:C9	RDF1-BCV+Mir	N/Asst'd	NR	4315

The `symbcv` command associates the local BCV devices with the device group; the `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
# symbcv -g symrep associateall dev -range 182:19A
```

The `symdev list` command with the `-bcv` option displays those BCV devices in the remote Symmetrix (sid 41) that are not already part of a device group (N/Asst'd) and which are free to be added to the device group. The display below has been edited to show those devices that are used in the example.

```
# symdev list -sid 41 -bcv
```

Symmetrix ID: 000187900041

Device Name		Directors		Device			
Sym	Physical	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)
0137	Not Visible	01C:0	02B:C8	BCV	N/Asst'd (M)	RW	12946
0138	Not Visible	01C:0	01A:C8	BCV	N/Asst'd (m)	RW	-
0139	Not Visible	01C:0	15A:C8	BCV	N/Asst'd (m)	RW	-
013A	Not Visible	01C:0	16A:CB	BCV	N/Asst'd (M)	RW	8631
013B	Not Visible	01C:0	02A:CB	BCV	N/Asst'd (m)	RW	-
013C	Not Visible	01C:0	01B:C7	BCV	N/Asst'd	RW	4315
013D	Not Visible	01C:0	15B:C7	BCV	N/Asst'd	RW	4315
013E	Not Visible	01C:0	16B:C0	BCV	N/Asst'd	RW	4315
013F	Not Visible	01C:0	02B:C0	BCV	N/Asst'd	RW	4315
0140	Not Visible	01C:0	01A:C0	BCV	N/Asst'd	RW	4315
0141	Not Visible	01C:0	15A:C0	BCV	N/Asst'd	RW	4315
0142	Not Visible	01C:0	16A:CD	BCV	N/Asst'd	RW	4315
0143	Not Visible	01C:0	02A:CD	BCV	N/Asst'd	RW	4315
0144	Not Visible	01C:0	01B:CB	BCV	N/Asst'd	RW	4315
0145	Not Visible	01C:0	15B:CB	BCV	N/Asst'd	RW	4315
0146	Not Visible	01C:0	16B:C6	BCV	N/Asst'd	RW	4315
0147	Not Visible	01C:0	02B:C6	BCV	N/Asst'd	RW	4315
0148	Not Visible	01C:0	01A:C2	BCV	N/Asst'd	RW	4315
0149	Not Visible	01C:0	15A:C2	BCV	N/Asst'd	RW	4315
014A	Not Visible	01C:0	16A:C1	BCV	N/Asst'd	RW	4315
014B	Not Visible	01C:0	02A:C1	BCV	N/Asst'd	RW	4315
014C	Not Visible	01C:0	01B:CD	BCV	N/Asst'd	RW	4315
014D	Not Visible	01C:0	15B:CD	BCV	N/Asst'd	RW	4315
014E	Not Visible	01C:0	16B:CA	BCV	N/Asst'd	RW	4315
014F	Not Visible	01C:0	02B:CA	BCV	N/Asst'd	RW	4315

The `symbcv` command with the `-rdf` and `-bcv` options associates the *remote* BCV devices with the device group.

```
# symbcv -g symrep associateall dev -range 137:14F -bcv -rdf
```

The `syndg show` command displays detailed group information about device group `symrep`. The group contains 22 local standard devices, 22 local R1 BCVs, and 22 remote BCVs.

```
# syndg show symrep
```

Group Name: symrep

```
Group Type           : REGULAR
Device Group in GNS  : No
Valid                : Yes
Symmetrix ID         : 000187900035
Group Creation Time  : Mon Nov 17 11:27:08 2003
Vendor ID            : EMC Corp
Application ID       : SYMCLI
```

```
Number of STD Devices in Group : 22
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 22
Number of Locally-associated VDEV's : 0
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 22
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 0
```

Standard (STD) Devices (22):

```
{
-----
LdevName          PdevName          Sym   Att. Sts   Cap
Dev              Dev              Dev
-----
DEV001            /dev/vx/rdmp/c15t2d2s2  0056 (M)  RW   12946
DEV002            /dev/vx/rdmp/c15t2d3s2  0059 (M)  RW   8631
DEV003            /dev/vx/rdmp/c15t2d4s2  005B      RW   4315
DEV004            /dev/vx/rdmp/c15t2d5s2  005C      RW   4315
DEV005            /dev/vx/rdmp/c15t2d6s2  005D      RW   4315
DEV006            /dev/vx/rdmp/c15t2d7s2  005E      RW   4315
DEV007            /dev/vx/rdmp/c15t2d8s2  005F      RW   4315
DEV008            /dev/vx/rdmp/c15t2d9s2  0060      RW   4315
DEV009            /dev/vx/rdmp/c15t2d10s2 0061      RW   4315
DEV010            /dev/vx/rdmp/c15t2d11s2 0062      RW   4315
DEV011            /dev/vx/rdmp/c15t2d12s2 0063      RW   4315
DEV012            /dev/vx/rdmp/c15t2d13s2 0064      RW   4315
DEV013            /dev/vx/rdmp/c15t2d14s2 0065      RW   4315
DEV014            /dev/vx/rdmp/c15t2d15s2 0066      RW   4315
DEV015            /dev/vx/rdmp/c15t2d16s2 0067      RW   4315
DEV016            /dev/vx/rdmp/c15t2d17s2 0068      RW   4315
DEV017            /dev/vx/rdmp/c15t2d18s2 0069      RW   4315
DEV018            /dev/vx/rdmp/c15t2d19s2 006A      RW   4315
DEV019            /dev/vx/rdmp/c15t2d20s2 006B      RW   4315
DEV020            /dev/vx/rdmp/c15t2d21s2 006C      RW   4315
DEV021            /dev/vx/rdmp/c15t2d22s2 006D      RW   4315
DEV022            /dev/vx/rdmp/c15t2d23s2 006E      RW   4315
```

}

BCV Devices Locally-associated (22):

{

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
BCV001	N/A	0182	(M)	NR	12946
BCV002	N/A	0185	(M)	NR	8631
BCV003	N/A	0187		NR	4315
BCV004	N/A	0188		NR	4315
BCV005	N/A	0189		NR	4315
BCV006	N/A	018A		NR	4315
BCV007	N/A	018B		NR	4315
BCV008	N/A	018C		NR	4315
BCV009	N/A	018D		NR	4315
BCV010	N/A	018E		NR	4315
BCV011	N/A	018F		NR	4315
BCV012	N/A	0190		NR	4315
BCV013	N/A	0191		NR	4315
BCV014	N/A	0192		NR	4315
BCV015	N/A	0193		NR	4315
BCV016	N/A	0194		NR	4315
BCV017	N/A	0195		NR	4315
BCV018	N/A	0196		NR	4315
BCV019	N/A	0197		NR	4315
BCV020	N/A	0198		NR	4315
BCV021	N/A	0199		NR	4315
BCV022	N/A	019A		NR	4315

}

BCV Devices Remotely-associated (BCV RDF) (22):

{

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
BRBCV001	N/A	0137	(M)	RW	12946
BRBCV002	N/A	013A	(M)	RW	8631
BRBCV003	N/A	013C		RW	4315
BRBCV004	N/A	013D		RW	4315
BRBCV005	N/A	013E		RW	4315
BRBCV006	N/A	013F		RW	4315
BRBCV007	N/A	0140		RW	4315
BRBCV008	N/A	0141		RW	4315
BRBCV009	N/A	0142		RW	4315
BRBCV010	N/A	0143		RW	4315
BRBCV011	N/A	0144		RW	4315
BRBCV012	N/A	0145		RW	4315
BRBCV013	N/A	0146		RW	4315
BRBCV014	N/A	0147		RW	4315
BRBCV015	N/A	0148		RW	4315
BRBCV017	N/A	014A		RW	4315
BRBCV018	N/A	014B		RW	4315
BRBCV019	N/A	014C		RW	4315
BRBCV020	N/A	014D		RW	4315

BRBCV021	N/A	014E	RW	4315
BRBCV022	N/A	014F	RW	4315
}				

Device Group BCV RDF Information

```

{
  RDF Type                               : R1
  RDF (RA) Group Number                   : 1                (00)

  Remote Symmetrix ID                     : 000187900041

  R2 Device Is Larger Than The R1 Device : False

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

  RDF Device Domino                       : Disabled

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

  Device RDF Status                       : Ready             (RW)

  Device RA Status                        : Ready             (RW)
  Device Link Status                      : Not Ready         (NR)

  Device Suspend State                    : Offline
  Device Consistency State                : Disabled
  RDF R2 Not Ready If Invalid             : Enabled

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

  RDF Pair State ( R1 <- -> R2 )         : Suspended

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

```

The following command illustrates the use of the vi text editor to create a text file named symrep.opt. As was done here, you can enter into the file those parameters and values that specify the single hop configuration and define copy cycle parameters for use during the symreplicate session: one cycle with a duration of 10 minutes. The `CYCLE_OVERFLOW` value of `NEXT` has no relevance here in a setup that has only one copy cycle, but this value will play a role later when the file is edited to have two copy cycles.

```
# vi symrep.opt
```

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=1
```

The `symreplicate setup` command performs the setup required to begin a replicate session. The difference between this command or using the `-setup` option with the `symreplicate start` command is that the latter will cycle as many times as you have specified in your `symreplicate` options file, whereas the `setup` command will cycle just once. If you have various “wait” options in the options file, the setup honors them.

```
# symreplicate -g symrep setup -optimize -options symrep.opt -foreground -nop
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Setting up local BCV pairs...
```

```
Optimizing Local BCV pairs...
```

```
Waiting for local BCV synchronization...
```

```
Splitting local BCV pairs...
```

```
Incrementally establishing RDF pairs...
```

```
Setting up remote BCV pairs...
```

```
Optimizing remote BCV pairs...
```

```
Incrementally establishing remote BCV pairs...
```

```
Waiting for remote device synchronization...
```

```
Waiting for RDF synchronization...
```

```
Splitting remote BCV pairs...
```

```
Incrementally establishing local BCV pairs...
```

```
Setup complete; exiting symreplicate...
```

The following command runs a symreplicate session in the foreground so that the resulting output display illustrates the various steps involved in completing one copy cycle. Note that the symrep.opt file specified on the command line tells SYMCLI what copy cycle parameters to employ during the session. The `-consistent` option performs a consistent split of the local BCV pairs during the cycle.

```
# symreplicate -g symrep start -options symrep.opt -foreground -consistent -nop
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Waiting for local BCV synchronization...
```

```
Splitting local BCV pairs...
```

```
Incrementally establishing RDF pairs...
```

```
Waiting for RDF synchronization...
```

```
Suspending RDF connection...
```

```
Waiting for local BCV pairs to split...
```

```
Incrementally establishing local BCV pairs...
```

```
Incrementally establishing remote BCV pairs...
```

```
Waiting for remote device synchronization...
```

```
Splitting remote BCV pairs...
```

```
1 cycle(s) complete; exiting symreplicate...
```

This `symreplicate` command runs one copy cycle in the background (the default when the `-foreground` option is omitted). A subsequent `symreplicate query` will check the status of the cycle being processed in the background.

```
# symreplicate -g symrep start -options symrep.opt -noprompt
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
symreplicate process launched.
```

The `symreplicate query` command checks the status of the copy cycle being processed in the background. SYMCLI provides an updated display every five seconds. Only a representative sample of the update displays is shown below.

```
# symreplicate -g symrep query -i 5
```

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Waiting for next cycle	0 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing RDF pairs	0 m	1	1

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Active	Establishing local and remote	0 m	1	1

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate Hop Type	Status	Step	Cycle Period	Current Cycle	Max Cycles
SINGLE	Completed	Complete	0 m	1	1

The following command uses the `vi` text editor again to edit the text file named `symrep.opt`. As was done here, you can edit parameter values that affect the `symreplicate` session. By changing the number of copy cycles from one to two, the `CYCLE_OVERFLOW` value of `NEXT` becomes relevant. If the first copy cycle lasts longer than its 10-minute time schedule, the second copy cycle will begin at the next scheduled start. For example, if the first copy cycle overflows to 15 minutes, the second cycle begins at the 20-minute mark. If the first copy cycle overflows to 35 minutes, the second cycle begins at the 40-minute mark.

```
# vi symrep.opt
```

```
SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=2
```

This `symreplicate start` command runs the two-cycle session in the background.

```
# symreplicate -g symrep start -options symrep.opt -noprompt
```

Checking for valid group configuration...

Checking for valid initial group state...

The `symreplicate query` command checks the status of the copy cycles being processed in the background.

```
# symreplicate -g symrep query
```

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Establishing RDF pairs		10 m	1	2

The `symreplicate stop` command stops the current replicate session. The `-step` option causes the stop to occur after the current execution step completes. Omitting `-step` would stop the session at the end of a complete copy cycle.

```
# symreplicate -g symrep stop -step -noprompt
```

Stop operation underway.

The `symreplicate query` command checks the status of the session. The display indicates that the session stopped at the step for establishing the RDF pairs.

```
# symreplicate -g symrep query
```

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Stopped	Establishing RDF pairs		10 m	1	2

The `symreplicate restart` command resumes the copy cycle at the step where the session stopped. Specifying the options file again on `restart` is not required unless you changed the file while the session was stopped. Although the example made no changes to the options file here, the options file is specified again for consistency.

```
# symreplicate -g symrep restart -options symrep.opt -noprompt
```

symreplicate process launched.

The `symreplicate query` command checks the status of the copy cycle every five seconds and provides an updated display. Note that the cycle resumes where it was in the sequence of steps when the session stopped. Only a representative sample of the update displays is shown below.

```
# symreplicate -g symrep query -i 5
```

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Establishing RDF pairs		10 m	1	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Establishing local and remote		10 m	1	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Waiting for next cycle		10 m	1	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Establishing RDF pairs		10 m	2	2

```
Device Group (DG) Name      : symrep
DG's Symmetrix ID          : 000187900035
Remote Symmetrix ID        : 000187900041
```

Replicate				Cycle	Current	Max
Hop Type	Status	Step		Period	Cycle	Cycles
SINGLE	Active	Establishing local and remote		10 m	2	2

Device Group (DG) Name : symrep
DG's Symmetrix ID : 000187900035
Remote Symmetrix ID : 000187900041

Replicate			Cycle	Current	Max
Hop Type	Status	Step	Period	Cycle	Cycles
-----	-----	-----	-----	-----	-----
SINGLE	Completed	Completed	10 m	2	2

Example 2: SRDF/AR Multi-Hop Configuration with BCVs at Hop 2

This example is performed using Solutions Enabler version 5.4. The hardware setup consists of a Solaris host connected to a source Symmetrix (sid 79). All commands are issued from the source-side host. The example uses the following devices to create the multi-hop environment and shows how to set up the correct pair states for automated data replication in this environment:

- Source Symmetrix (sid 79): R1 devices CEE-CFD
- Hop 1 Symmetrix (sid 67): R2 devices EA2-EB1; R1 BCV devices E92-EA1
- Hop 2 Symmetrix (sid 01): R2 devices 318-327; BCV devices 328-337

The `symrdf list` command shows the local view of the source R1 devices (SymDev), their target R2 mirror devices (RDev), and their current SRDF pair state. The ellipsis (.....) represents truncated output.

```
# symrdf list -r1 -sid 79
```

```
Symmetrix ID: 000187700079
```

Local Device View

		STATUS			MODES		RDF S T A T E S				
Sym		RDF	-----			-----	R1 Inv	R2 Inv	-----		
Dev	RDev	Typ:G	SA	RA	LNK	MDA	Tracks	Tracks	Dev	RDev	Pair
----	----	-----	----	----	----	-----	-----	-----	----	----	-----
0CEE	0EA2	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CEF	0EA3	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF0	0EA4	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF1	0EA5	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF2	0EA6	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF3	0EA7	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF4	0EA8	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF5	0EA9	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF6	0EAA	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF7	0EAB	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF8	0EAC	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CF9	0EAD	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CFA	0EAE	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CFB	0EAF	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CFC	0EB0	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0CFD	0EB1	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized

The `symdg create` command creates a device group (symrep). The `symld addall` command adds the RDF standard devices from local Symmetrix 000187700079 to the group, using the `-range` option to limit the selections to those devices between CEE and CFD. The `symbcv` command with the `-rdf` option associates a range of remote BCV devices on Hop 1 with the device group. The `symbcv` command with the `-rrdf` option associates a range of remote BCV devices on Hop 2 with the device group.

```
# symdg create symrep -type rdf1
# symld -g symrep addall dev -range CEE:CFD -sid 79
# symbcv -g symrep -rdf associateall dev -range E92:EA1
# symbcv -g symrep -rrdf associateall dev -range 328:337
```

The following command illustrates the use of the vi text editor to create a text file named rep_opt.txt. As was done here, you can enter into the file those parameters and values that specify the multi-hop configuration that uses the Hop 2 BCVs (USE_FINAL_BCV=TRUE) and define copy cycle parameters for use during the symreplicate session. The CYCLE_OVERFLOW value of NEXT has no relevance here in a setup that has only one copy cycle, but this value can play a role later if the file is edited to have more than one copy cycle.

```
# vi rep_opt.txt
```

```
SYMCLI_REPLICATE_HOP_TYPE=MULTI
SYMCLI_REPLICATE_USE_FINAL_BCV=TRUE
SYMCLI_REPLICATE_CYCLE=0
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=1
```

The symreplicate start command with the -setup option sets up the required pair states, and if successful, begins the symreplicate session. This command will cycle as many times as you have specified in your symreplicate options file (the symreplicate setup command cycles just once).

```
# symreplicate -g symrep start -setup -optimize -options rep_opt.txt -foreground \
                                     -noprompt
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Setting up local RDF pairs...
```

```
Setting up first hop BCV pairs...
```

```
Optimizing first hop BCV pairs...
```

```
Waiting for first hop BCV device synchronization...
```

```
Splitting first hop BCV pairs...
```

```
Incrementally establishing remote RDF pairs...
```

```
Setting up second hop BCV pairs...
```

```
Optimizing second hop BCV pairs...
```

```
Incrementally establishing second hop BCV pairs...
```

```
Waiting for second hop BCV synchronization...
```

```
Waiting for remote RDF pair synchronization...
```

```
Splitting second hop BCV pairs...
```

```
Incrementally establishing first hop BCV pairs...
```

```
Setup is complete...
```

```
1 cycle(s) complete; exiting symreplicate...
```

Example 3: Setting Up an SRDF/AR Single-Hop Configuration Using a Composite Group

This example is performed using Solutions Enabler version 5.4. The hardware setup illustrates a single-hop configuration in which the source devices span three Symmetrix arrays (SIDs 35, 43, and 60). A composite group is defined on a host connected to these three Symmetrix arrays. The devices include standard devices and R1 BCV devices from the local Symmetrix arrays, as well as BCVs from the remote Symmetrix arrays.

The `symcg create` command creates a Regular type composite group named `single-hop`.

```
# symcg create single-hop -type regular
```

The following `symcg` commands add to the composite group a range of standard devices from each of the three local source Symmetrix arrays.

```
# symcg -cg single-hop addall dev -range 56:6E -sid 35
# symcg -cg single-hop addall dev -range 61:79 -sid 43
# symcg -cg single-hop addall dev -range 14:27 -sid 60
```

The following `symbcv` commands associate with the composite group a range of R1 BCV devices from each of the three local source Symmetrix arrays.

```
# symbcv -cg single-hop associateall dev -range 182:19A -sid 35
# symbcv -cg single-hop associateall dev -range 142:15A -sid 43
# symbcv -cg single-hop associateall dev -range 3B6:3C9 -sid 60
```

The following `symbcv` commands with the `-rdf` option associate a range of BCV devices from each of the *remote* Symmetrix arrays. The `-bcv` option specifies that the source devices are local R1 BCV devices. If there is more than one RDF group on a local Symmetrix array, you must include the RDF group number of the local source devices (the group number of the R1 BCVs). Specifying the RDF group number creates each R1/R2 pairing as well as the RDF link for that pair.

```
# symbcv -cg single-hop associateall dev -range 137:14F -bcv -rdf -sid 35 -rdfg 1
# symbcv -cg single-hop associateall dev -range 12A:142 -bcv -rdf -sid 43 -rdfg 1
# symbcv -cg single-hop associateall dev -range 21C:22F -bcv -rdf -sid 60 -rdfg 1
```

The `symcg list` command displays a list of composite groups defined on this host.

```
# symcg list
```

C O M P O S I T E G R O U P S								
Name	Type	Valid	Number of		Number of			
			Symms	RAGs	Devs	BCVs	VDEVs	
single-hop	REGULAR	Yes	3	3	64	128	0	

The `symcg show` command displays detailed configuration and status information about the composite group.

symcg show single-hop

Composite Group Name: single-hop

```
Composite Group Type      : REGULAR
Valid                    : Yes
CG in PowerPath          : No
CG in GNS                : No
```

```
Number of RDF (RA) Groups      : 3
Number of STD Devices         : 64
Number of BCV's (Locally-associated) : 64
Number of VDEV's (Locally-associated) : 0
Number of RBCV's (Remotely-associated STD-RDF) : 0
Number of BRBCV's (Remotely-associated BCV-RDF) : 64
Number of RRBCV's (Remotely-associated RBCV) : 0
```

Number of Symmetrix Units (3):

```
{
  1) Symmetrix ID              : 000187900035
     Microcode Version         : 5670
     Number of STD Devices     : 22
     Number of BCV's (Locally-associated) : 22
     Number of VDEV's (Locally-associated) : 0
     Number of RBCV's (Remotely-associated STD_RDF) : 0
     Number of BRBCV's (Remotely-associated BCV-RDF) : 22
     Number of RRBCV's (Remotely-associated RBCV) : 0
```

Number of RDF (RA) Groups (1):

```
{
  1) RDF (RA) Group Number : 1 (00)
     Remote Symmetrix ID   : 000187900041
     Microcode Version     : 5670
```

BCV's (Locally-associated) (22):

```
{
-----
PdevName          Sym  Device      Consistency   Cap
                  Dev  Config      State         (MB)
-----
N/A                0182 RDF1-BCV+Mir Disabled     12946
N/A                0185 RDF1-BCV+Mir Disabled      8631
N/A                0187 RDF1-BCV+Mir Disabled      4315
N/A                0188 RDF1-BCV+Mir Disabled      4315
N/A                0189 RDF1-BCV+Mir Disabled      4315
N/A                018A RDF1-BCV+Mir Disabled      4315
N/A                018B RDF1-BCV+Mir Disabled      4315
N/A                018C RDF1-BCV+Mir Disabled      4315
N/A                018D RDF1-BCV+Mir Disabled      4315
N/A                018E RDF1-BCV+Mir Disabled      4315
N/A                018F RDF1-BCV+Mir Disabled      4315
```

```

N/A          0190 RDF1-BCV+Mir Disabled 4315
N/A          0191 RDF1-BCV+Mir Disabled 4315
N/A          0192 RDF1-BCV+Mir Disabled 4315
N/A          0193 RDF1-BCV+Mir Disabled 4315
N/A          0194 RDF1-BCV+Mir Disabled 4315
N/A          0195 RDF1-BCV+Mir Disabled 4315
N/A          0196 RDF1-BCV+Mir Disabled 4315
N/A          0197 RDF1-BCV+Mir Disabled 4315
N/A          0198 RDF1-BCV+Mir Disabled 4315
N/A          0199 RDF1-BCV+Mir Disabled 4315
N/A          019A RDF1-BCV+Mir Disabled 4315
}

```

BRBCV's (Remotely-associated BCV-RDF) (22):

```

{
-----
PdevName          Sym  Device      Consistency    Cap
                   Dev  Config      State          (MB)
-----
N/A                0137 BCV          N/A            12946
N/A                013A BCV          N/A            8631
N/A                013C BCV          N/A            4315
N/A                013D BCV          N/A            4315
N/A                013E BCV          N/A            4315
N/A                013F BCV          N/A            4315
N/A                0140 BCV          N/A            4315
N/A                0141 BCV          N/A            4315
N/A                0142 BCV          N/A            4315
N/A                0143 BCV          N/A            4315
N/A                0144 BCV          N/A            4315
N/A                0145 BCV          N/A            4315
N/A                0146 BCV          N/A            4315
N/A                0147 BCV          N/A            4315
N/A                0148 BCV          N/A            4315
N/A                0149 BCV          N/A            4315
N/A                014A BCV          N/A            4315
N/A                014B BCV          N/A            4315
N/A                014C BCV          N/A            4315
N/A                014D BCV          N/A            4315
N/A                014E BCV          N/A            4315
N/A                014F BCV          N/A            4315
}

```

}

STD Devices (non-RDF) (22):

```

{
-----
PdevName          Sym  Device      Consistency    Cap
                   Dev  Config      State          (MB)
-----
/dev/vx/rdmp/c15t2d2s2 0056 2-Way Mir    N/A            12946
/dev/vx/rdmp/c15t2d3s2 0059 2-Way Mir    N/A            8631
/dev/vx/rdmp/c15t2d4s2 005B 2-Way Mir    N/A            4315
/dev/vx/rdmp/c15t2d5s2 005C 2-Way Mir    N/A            4315
/dev/vx/rdmp/c15t2d6s2 005D 2-Way Mir    N/A            4315
/dev/vx/rdmp/c15t2d7s2 005E 2-Way Mir    N/A            4315
/dev/vx/rdmp/c15t2d8s2 005F 2-Way Mir    N/A            4315
}

```

```

/dev/vx/rdmp/c15t2d9s2 0060 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d10s2 0061 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d11s2 0062 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d12s2 0063 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d13s2 0064 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d14s2 0065 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d15s2 0066 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d16s2 0067 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d17s2 0068 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d18s2 0069 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d19s2 006A 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d20s2 006B 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d21s2 006C 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d22s2 006D 2-Way Mir N/A 4315
/dev/vx/rdmp/c15t2d23s2 006E 2-Way Mir N/A 4315
}

```

```

2) Symmetrix ID : 000000003143
Microcode Version : 5267
Number of STD Devices : 22
Number of BCV's (Locally-associated) : 22
Number of VDEV's (Locally-associated) : 0
Number of RBCV's (Remotely-associated STD_RDF) : 0
Number of BRBCV's (Remotely-associated BCV-RDF) : 22
Number of RRBCV's (Remotely-associated RBCV) : 0

```

Number of RDF (RA) Groups (1):

{

```

1) RDF (RA) Group Number : 1 (A)
Remote Symmetrix ID : 000000003156
Microcode Version : 5267

```

BCV's (Locally-associated) (22):

{

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0142	RDF1-BCV	Disabled	12946
N/A	0145	RDF1-BCV	Disabled	8631
N/A	0147	RDF1-BCV	Disabled	4315
N/A	0148	RDF1-BCV	Disabled	4315
N/A	0149	RDF1-BCV	Disabled	4315
N/A	014A	RDF1-BCV	Disabled	4315
N/A	014B	RDF1-BCV	Disabled	4315
N/A	014C	RDF1-BCV	Disabled	4315
N/A	014D	RDF1-BCV	Disabled	4315
N/A	014E	RDF1-BCV	Disabled	4315
N/A	014F	RDF1-BCV	Disabled	4315
N/A	0150	RDF1-BCV	Disabled	4315
N/A	0151	RDF1-BCV	Disabled	4315
N/A	0152	RDF1-BCV	Disabled	4315
N/A	0153	RDF1-BCV	Disabled	4315
N/A	0154	RDF1-BCV	Disabled	4315
N/A	0155	RDF1-BCV	Disabled	4315
N/A	0156	RDF1-BCV	Disabled	4315

```

N/A                0157 RDF1-BCV      Disabled      4315
N/A                0158 RDF1-BCV      Disabled      4315
N/A                0159 RDF1-BCV      Disabled      4315
N/A                015A RDF1-BCV      Disabled      4315
}

```

BRBCV's (Remotely-associated BCV-RDF) (22):

```

{
-----
PdevName                Sym  Device      Consistency  Cap
                        Dev  Config      State        (MB)
-----
N/A                    012A BCV          N/A          12946
N/A                    012D BCV          N/A          8631
N/A                    012F BCV          N/A          4315
N/A                    0130 BCV          N/A          4315
N/A                    0131 BCV          N/A          4315
N/A                    0132 BCV          N/A          4315
N/A                    0133 BCV          N/A          4315
N/A                    0134 BCV          N/A          4315
N/A                    0135 BCV          N/A          4315
N/A                    0136 BCV          N/A          4315
N/A                    0137 BCV          N/A          4315
N/A                    0138 BCV          N/A          4315
N/A                    0139 BCV          N/A          4315
N/A                    013A BCV          N/A          4315
N/A                    013B BCV          N/A          4315
N/A                    013C BCV          N/A          4315
N/A                    013D BCV          N/A          4315
N/A                    013E BCV          N/A          4315
N/A                    013F BCV          N/A          4315
N/A                    0140 BCV          N/A          4315
N/A                    0141 BCV          N/A          4315
N/A                    0142 BCV          N/A          4315
}
}

```

STD Devices (non-RDF) (22):

```

{
-----
PdevName                Sym  Device      Consistency  Cap
                        Dev  Config      State        (MB)
-----
/dev/vx/rdmp/c15t1d2s2  0061 2-Way Mir    N/A          12946
/dev/vx/rdmp/c15t1d3s2  0064 2-Way Mir    N/A          8631
/dev/vx/rdmp/c15t1d4s2  0066 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d5s2  0067 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d6s2  0068 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d7s2  0069 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d8s2  006A 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d9s2  006B 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d10s2 006C 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d11s2 006D 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d12s2 006E 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d13s2 006F 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d14s2 0070 2-Way Mir    N/A          4315
/dev/vx/rdmp/c15t1d15s2 0071 2-Way Mir    N/A          4315
}

```

```

/dev/vx/rdmp/c15t1d16s2 0072 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d17s2 0073 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d18s2 0074 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d19s2 0075 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d20s2 0076 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d21s2 0077 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d22s2 0078 2-Way Mir      N/A      4315
/dev/vx/rdmp/c15t1d23s2 0079 2-Way Mir      N/A      4315
}

```

```

3) Symmetrix ID           : 000184500160
   Microcode Version       : 5568
   Number of STD Devices   : 20
   Number of BCV's (Locally-associated) : 20
   Number of VDEV's (Locally-associated) : 0
   Number of RBCV's (Remotely-associated STD RDF) : 0
   Number of BRBCV's (Remotely-associated BCV-RDF) : 20
   Number of RRBCV's (Remotely-associated RBCV) : 0

```

Number of RDF (RA) Groups (1):

```

{
  1) RDF (RA) Group Number : 1          (A)
     Remote Symmetrix ID   : 000184502898
     Microcode Version     : 5568

```

BCV's (Locally-associated) (20):

```

{
-----
PdevName          Sym  Device      Consistency   Cap
                  Dev  Config      State          (MB)
-----
N/A                03B6 RDF1-BCV+Mir Disabled      4315
N/A                03B7 RDF1-BCV+Mir Disabled      4315
N/A                03B8 RDF1-BCV+Mir Disabled      4315
N/A                03B9 RDF1-BCV+Mir Disabled      4315
N/A                03BA RDF1-BCV+Mir Disabled      4315
N/A                03BB RDF1-BCV+Mir Disabled      4315
N/A                03BC RDF1-BCV+Mir Disabled      4315
N/A                03BD RDF1-BCV+Mir Disabled      4315
N/A                03BE RDF1-BCV+Mir Disabled      4315
N/A                03BF RDF1-BCV+Mir Disabled      4315
N/A                03C0 RDF1-BCV+Mir Disabled      4315
N/A                03C1 RDF1-BCV+Mir Disabled      4315
N/A                03C2 RDF1-BCV+Mir Disabled      4315
N/A                03C3 RDF1-BCV+Mir Disabled      4315
N/A                03C4 RDF1-BCV+Mir Disabled      4315
N/A                03C5 RDF1-BCV+Mir Disabled      4315
N/A                03C6 RDF1-BCV+Mir Disabled      4315
N/A                03C7 RDF1-BCV+Mir Disabled      4315
N/A                03C8 RDF1-BCV+Mir Disabled      4315
N/A                03C9 RDF1-BCV+Mir Disabled      4315
}

```

BRBCV's (Remotely-associated BCV-RDF) (20):

```

{
-----

```

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	021C	2-Way BCV Mir	N/A	4315
N/A	021D	2-Way BCV Mir	N/A	4315
N/A	021E	2-Way BCV Mir	N/A	4315
N/A	021F	2-Way BCV Mir	N/A	4315
N/A	0220	2-Way BCV Mir	N/A	4315
N/A	0221	2-Way BCV Mir	N/A	4315
N/A	0222	2-Way BCV Mir	N/A	4315
N/A	0223	2-Way BCV Mir	N/A	4315
N/A	0224	2-Way BCV Mir	N/A	4315
N/A	0225	2-Way BCV Mir	N/A	4315
N/A	0226	2-Way BCV Mir	N/A	4315
N/A	0227	2-Way BCV Mir	N/A	4315
N/A	0228	2-Way BCV Mir	N/A	4315
N/A	0229	2-Way BCV Mir	N/A	4315
N/A	022A	2-Way BCV Mir	N/A	4315
N/A	022B	2-Way BCV Mir	N/A	4315
N/A	022C	2-Way BCV Mir	N/A	4315
N/A	022D	2-Way BCV Mir	N/A	4315
N/A	022E	2-Way BCV Mir	N/A	4315
N/A	022F	2-Way BCV Mir	N/A	4315
}				
}				

STD Devices (non-RDF) (20):

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
/dev/rdisk/emcpower38c	0014	2-Way Mir	N/A	4315
/dev/rdisk/emcpower39c	0015	2-Way Mir	N/A	4315
/dev/rdisk/emcpower40c	0016	2-Way Mir	N/A	4315
/dev/rdisk/emcpower41c	0017	2-Way Mir	N/A	4315
/dev/rdisk/emcpower42c	0018	2-Way Mir	N/A	4315
/dev/rdisk/emcpower43c	0019	2-Way Mir	N/A	4315
/dev/rdisk/emcpower44c	001A	2-Way Mir	N/A	4315
/dev/rdisk/emcpower45c	001B	2-Way Mir	N/A	4315
/dev/rdisk/emcpower46c	001C	2-Way Mir	N/A	4315
/dev/rdisk/emcpower47c	001D	2-Way Mir	N/A	4315
/dev/rdisk/emcpower48c	001E	2-Way Mir	N/A	4315
/dev/rdisk/emcpower49c	001F	2-Way Mir	N/A	4315
/dev/rdisk/emcpower50c	0020	2-Way Mir	N/A	4315
/dev/rdisk/emcpower51c	0021	2-Way Mir	N/A	4315
/dev/rdisk/emcpower52c	0022	2-Way Mir	N/A	4315
/dev/rdisk/emcpower53c	0023	2-Way Mir	N/A	4315
/dev/rdisk/emcpower54c	0024	2-Way Mir	N/A	4315
/dev/rdisk/emcpower55c	0025	2-Way Mir	N/A	4315
/dev/rdisk/emcpower56c	0026	2-Way Mir	N/A	4315
/dev/rdisk/emcpower57c	0027	2-Way Mir	N/A	4315
}				
}				

The following command illustrates the use of the vi text editor to create a text file named sar.opt. The setup operation requires that the HOP_TYPE be defined. You define copy cycle parameters for use during the symreplicate session (for example, two cycles where the time from the beginning of the first cycle to the beginning of the next cycle should be 10 minutes).

```
# vi sar.opt

SYMCLI_REPLICATE_HOP_TYPE=SINGLE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=2
```

Similar to Example 1, the symreplicate setup command performs the setup here for all pairs in the composite group that spans three source Symmetrix arrays. This command results in the setup performing one cycle.

```
# symreplicate -cg single-hop setup -foreground -options sar.opt -noprompt
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Setting up local BCV pairs...
```

```
Waiting for local BCV synchronization...
```

```
Splitting local BCV pairs...
```

```
Incrementally establishing RDF pairs...
```

```
Setting up remote BCV pairs...
```

```
Incrementally establishing remote BCV pairs...
```

```
Waiting for remote device synchronization...
```

```
Waiting for RDF synchronization...
```

```
Splitting remote BCV pairs...
```

```
Incrementally establishing local BCV pairs...
```

```
Setup complete; exiting symreplicate...
```

The setup is complete. You can now perform symreplicate start for the composite group as was done for the device group in Example 1.

Example 4: Setting Up an SRDF/AR Multi-Hop Configuration Using a Composite Group

This example is performed using Solutions Enabler version 5.4. The hardware setup illustrates a multi-hop configuration in which the source devices span two Symmetrix arrays (SIDs 79 and 32). A composite group is defined on a Solaris host connected to these two Symmetrix arrays. The devices include R1 devices from the local Symmetrix arrays, as well as BCVs from on the remote Hop-1 and Hop-2 Symmetrix arrays.

The `symcg create` command creates an RDF1 type composite group named multi-hop.

```
# symcg create multi-hop -type rdf1
```

The following `symcg` commands add to the composite group a range of standard devices from each of the two local source Symmetrix arrays.

```
# symcg -cg multi-hop addall dev -range CEE:CFD -sid 79
# symcg -cg multi-hop addall dev -range 2:7 -sid 32
```

The following `symbcv` commands with the `-rdf` option associate a range of BCV devices from each of the *remote* Hop-1 Symmetrix arrays. If there is more than one RDF group on a local Symmetrix array, you must include the RDF group number of the local R1 source devices. Specifying the RDF group number creates the RDF link for each R1/R2 pair.

```
# symbcv -cg multi-hop associateall dev -range E92:EA1 -rdf -rdfg 64 -sid 79
# symbcv -cg multi-hop associateall dev -range 48:4D -rdf -rdfg 1 -sid 32
```

The following `symbcv` commands with the `-rrdf` option associate a range of BCV devices from each of the *remote* Hop-2 Symmetrix arrays. To define the path of the RDF link, you must always include the group number of the local source devices as specified with the `-rdfg` option.⁷

```
# symbcv -cg multi-hop associateall dev -range 328:337 -rrdf -rdfg 64 -sid 79
# symbcv -cg multi-hop associateall dev -range 30:35 -rrdf -rdfg 1 -sid 32
```

The `symcg list` command displays a list of composite groups defined on this host.

```
# symcg list
```

C O M P O S I T E G R O U P S								
Name	Type	Valid	Number of		Number of			
			Symms	RAGs	Devs	BCVs	VDEVs	
multi-hop	RDF1	Yes	2	2	22	44	0	

The `symcg show` command displays configuration and status information about the composite group.

```
# symcg show multi-hop
```

```
Composite Group Name: multi-hop
```

```
Composite Group Type      : RDF1
Valid                     : Yes
```

⁷ If you add the Hop 2 BCVs to the composite group before adding the Hop 1 BCVs, include the `-remote_rdfg` option also.

```
CG in PowerPath          : No
CG in GNS                 : No
```

```
Number of RDF (RA) Groups      :    2
Number of STD Devices          :   22
Number of BCV's (Locally-associated) :    0
Number of VDEV's (Locally-associated) :    0
Number of RBCV's (Remotely-associated STD-RDF) :   22
Number of BRBCV's (Remotely-associated BCV-RDF) :    0
Number of RRBCV's (Remotely-associated RBCV) :   22
```

```
Number of Symmetrix Units (2):
```

```
{
  1) Symmetrix ID          : 000187700079
     Microcode Version      : 5670
     Number of STD Devices  :   16
     Number of BCV's (Locally-associated) :    0
     Number of VDEV's (Locally-associated) :    0
     Number of RBCV's (Remotely-associated STD_RDF) :   16
     Number of BRBCV's (Remotely-associated BCV-RDF) :    0
     Number of RRBCV's (Remotely-associated RBCV) :   16
```

```
Number of RDF (RA) Groups (1):
```

```
{
  1) RDF (RA) Group Number : 64          (3F)
     Remote Symmetrix ID   : 000187700067
     Microcode Version     : 5670
```

```
STD Devices (16):
```

```
{
-----
PdevName          Sym  Device  Consistency  Cap
                  Dev  Config  State        (MB)
-----
/dev/vx/rdmp/c15t0d63s2 0CEE RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d64s2 0CEF RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d65s2 0CF0 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d66s2 0CF1 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d67s2 0CF2 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d68s2 0CF3 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d69s2 0CF4 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d70s2 0CF5 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d71s2 0CF6 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d72s2 0CF7 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d73s2 0CF8 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d74s2 0CF9 RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d75s2 0CFA RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d76s2 0CFB RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d77s2 0CFC RDF1+Mir  Disabled    4315
/dev/vx/rdmp/c15t0d78s2 0CFD RDF1+Mir  Disabled    4315
}
```

```
RBCV's (Remotely-associated STD-RDF) (16):
```

```
{
  Remote RDF (RA) Group Number : 52          (33)
```

Remote Remote Symmetrix ID : 000000006201
 Microcode Version : 5670

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0E92	RDF1-BCV	Disabled	4315
N/A	0E93	RDF1-BCV	Disabled	4315
N/A	0E94	RDF1-BCV	Disabled	4315
N/A	0E95	RDF1-BCV	Disabled	4315
N/A	0E96	RDF1-BCV	Disabled	4315
N/A	0E97	RDF1-BCV	Disabled	4315
N/A	0E98	RDF1-BCV	Disabled	4315
N/A	0E99	RDF1-BCV	Disabled	4315
N/A	0E9A	RDF1-BCV	Disabled	4315
N/A	0E9B	RDF1-BCV	Disabled	4315
N/A	0E9C	RDF1-BCV	Disabled	4315
N/A	0E9D	RDF1-BCV	Disabled	4315
N/A	0E9E	RDF1-BCV	Disabled	4315
N/A	0E9F	RDF1-BCV	Disabled	4315
N/A	0EA0	RDF1-BCV	Disabled	4315
N/A	0EA1	RDF1-BCV	Disabled	4315

RRBCV's (Remotely-associated RBCV) (16):

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0328	BCV	N/A	4315
N/A	0329	BCV	N/A	4315
N/A	032A	BCV	N/A	4315
N/A	032B	BCV	N/A	4315
N/A	032C	BCV	N/A	4315
N/A	032D	BCV	N/A	4315
N/A	032E	BCV	N/A	4315
N/A	032F	BCV	N/A	4315
N/A	0330	BCV	N/A	4315
N/A	0331	BCV	N/A	4315
N/A	0332	BCV	N/A	4315
N/A	0333	BCV	N/A	4315
N/A	0334	BCV	N/A	4315
N/A	0335	BCV	N/A	4315
N/A	0336	BCV	N/A	4315
N/A	0337	BCV	N/A	4315

2) Symmetrix ID : 000000005232
 Microcode Version : 5568
 Number of STD Devices : 6
 Number of BCV's (Locally-associated) : 0
 Number of VDEV's (Locally-associated) : 0
 Number of RBCV's (Remotely-associated STD_RDF) : 6
 Number of BRBCV's (Remotely-associated BCV-RDF) : 0

Number of RRBCV's (Remotely-associated RBCV) : 6

Number of RDF (RA) Groups (1):

{

1) RDF (RA) Group Number : 1 (A)
 Remote Symmetrix ID : 000000005231
 Microcode Version : 5568

STD Devices (6):

{

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0002	RDF1	Disabled	1031
N/A	0003	RDF1	Disabled	1031
N/A	0004	RDF1	Disabled	1031
N/A	0005	RDF1	Disabled	1031
N/A	0006	RDF1	Disabled	1031
N/A	0007	RDF1	Disabled	1031

RBCV's (Remotely-associated STD-RDF) (6):

{

Remote RDF (RA) Group Number : 2 (B)
 Remote Remote Symmetrix ID : 000000005233
 Microcode Version : 5568

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0048	RDF1-BCV	Disabled	1031
N/A	0049	RDF1-BCV	Disabled	1031
N/A	004A	RDF1-BCV	Disabled	1031
N/A	004B	RDF1-BCV	Disabled	1031
N/A	004C	RDF1-BCV	Disabled	1031
N/A	004D	RDF1-BCV	Disabled	1031

RRBCV's (Remotely-associated RBCV) (6):

{

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
N/A	0030	BCV	N/A	1031
N/A	0031	BCV	N/A	1031
N/A	0032	BCV	N/A	1031
N/A	0033	BCV	N/A	1031
N/A	0034	BCV	N/A	1031
N/A	0035	BCV	N/A	1031

}

}

}

The following command illustrates the use of the vi text editor to create a text file named 3-hop.opt. The setup operation requires that the HOP_TYPE be defined. By default, USE_FINAL_BCV is set to TRUE. You define copy cycle parameters for use during the symreplicate session (for example, two cycles where the time from the beginning of the first cycle to the beginning of the next cycle should be 10 minutes).

```
# vi 3-hop_opt.txt
```

```
SYMCLI_REPLICATE_HOP_TYPE=MULTI
SYMCLI_REPLICATE_USE_FINAL_BCV=TRUE
SYMCLI_REPLICATE_CYCLE=10
SYMCLI_REPLICATE_CYCLE_OVERFLOW=NEXT
SYMCLI_REPLICATE_NUM_CYCLES=2
```

Similar to Example 1, the symreplicate setup command performs the setup here for all pairs in the composite group that spans two source Symmetrix arrays. This command results in the setup performing one cycle. The -optimize flag is included to optimize BCV device pairings within each Symmetrix array.

```
# symreplicate -cg multi-hop setup -optimize -foreground -options 3-hop.opt -nop
```

```
Checking for valid group configuration...
```

```
Checking for valid initial group state...
```

```
Setting up local RDF pairs...
```

```
Setting up first hop BCV pairs...
```

```
Optimizing first hop BCV pairs...
```

```
Waiting for first hop BCV device synchronization...
```

```
Splitting first hop BCV pairs...
```

```
Incrementally establishing remote RDF pairs...
```

```
Setting up second hop BCV pairs...
```

```
Optimizing second hop BCV pairs...
```

```
Incrementally establishing second hop BCV pairs...
```

```
Waiting for second hop BCV synchronization...
```

```
Waiting for remote RDF pair synchronization...
```

```
Splitting second hop BCV pairs...
```

```
Incrementally establishing first hop BCV pairs...
```

```
Setup complete; exiting symreplicate...
```

The setup is complete. You can now perform symreplicate start for the composite group as was done for the device group in Example 2.

Example 5: Accessing Concurrent Non-SRDF/AR BCVs While Running SRDF/AR (Single-Hop Configuration)⁸

The hardware setup for this single-hop SRDF/AR configuration consists of a host connected to a local source Symmetrix (sid 505) and a remote target Symmetrix. All commands are issued from the source-side host and affect devices on the local Symmetrix. The example establishes a set of non-SRDF/AR BCVs that are concurrent with a set of SRDF/AR BCVs. Devices on the local Symmetrix are:

- SRDF/AR standard devices 012E and 012F
- SRDF/AR BCV devices 04CE and 04CF
- Non-SRDF/AR BCV devices 04CC and 04CD

The SRDF/AR device group (sar) and the creation of the single-hop environment have already been set up. The SRDF/AR device group was created using the following commands:

```
symdg create sar
symld -g sar addall dev -range 12E:12F
symbcv -g sar associateall dev -range 4CE:4CF
symbcv -g sar associateall dev -range 30C:30D -bcv -rdf
```

A device file named devfile was created to define the following non-SRDF/AR BCV pairs:

```
012E 04CC
012F 04CD
```

The following commands were used to fully establish and split the non-SRDF/AR BCV pairs in the device file:

```
symmir -f devfile -sid 505 establish -full -noprompt
symmir -f devfile -sid 505 split -noprompt
```

The SRDF/AR devices were set up for a SRDF/AR copy cycle using the following commands that operate on the SRDF/AR devices in the device group named sar. (Beginning with Solutions Enabler version 5.4, you can perform these steps automatically with the `symreplicate setup` command.)

```
symmir -g sar establish -full -exact -noprompt
symmir -g sar split -noprompt
symrdf -g sar -bcv establish -noprompt
symrdf -g sar -bcv split -noprompt
symmir -g sar establish -full -exact -bcv -rdf -noprompt
symmir -g sar split -bcv -rdf -noprompt
symmir -g sar establish -noprompt
```

If a concurrent BCV setup exists when a SRDF/AR copy cycle begins, SRDF/AR will use the last BCV pair that was established, regardless of whether it was the SRDF/AR BCV pair or the non-SRDF/AR BCV pair. *Prior to starting SRDF/AR, you need to make sure that the last BCV pairs that were established were the SRDF/AR BCV pairs.* In the preceding sequence, the last BCV pairs established were the SRDF/AR BCV pairs. Thus, the required setup sequence has been performed.

⁸ The ability to access concurrent BCVs while running SDRD/AR begins with Solutions Enabler version 5.2 and Enginuity version 5568.

The following `symmir query` command with the `-multi` option examines the relationship and status of the concurrent BCV pairs. The SRDF/AR pairs are devices 012E/04CE and 012F/04CF. The non-SRDF/AR pairs are devices 012E/04CC and 012F/04CD. The non-SRDF/AR pairs are displayed even though they are *not* associated with the device group (indicated by the “N/A” and the absence of the *). All pairs are in the Synchronized state.

```
# symmir -g sar query -multi
```

```
Device Group (DG) Name: sar
DG's Type              : REGULAR
DG's Symmetrix ID      : 000185500505
```

Standard Device			BCV Device			State
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...

Splitting remote BCV pairs...

1 cycle(s) complete; exiting symreplicate...

Example 6: Accessing Concurrent Non-SRDF/AR BCVs While Running SRDF/AR (Multi-Hop Configuration)⁹

The hardware setup for this multi-hop 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 multi-hop environment have already been set up. The example establishes on the target Symmetrix a set of non-SRDF/AR BCVs that are concurrent with a set of SRDF/AR BCVs there. Devices on the target (Hop 2) Symmetrix are:

- SRDF/AR standard devices 0001–0005
- SRDF/AR BCV devices 0031–0035
- Non-SRDF/AR BCV devices 0043–0047

Although the 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 array (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`.¹⁰

```
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			0	
MB(s)		0.0			0.0	

Legend:

(*): The paired BCV device is associated with this group.

⁹ The ability to access concurrent BCVs while running SDRD/AR begins with Solutions Enabler version 5.2 and Enginuity version 5568.

¹⁰ 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 `symdg create` command creates an R2 type device group (mbcv). The `symld addall` command adds the SRDF/AR R2 devices to the group, using the `-range` option to limit the selections to those devices between 0001 and 0005. The `symbcv` command associates the non-SRDF/AR BCVs with the device group, using the `-range` option with the `associateall` action to limit the selection to those BCVs that are within the specified range (0043 through 0047).

```
tgt# symdg create mbcv -type rdf2
tgt# symld -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	SyncInProgress
DEV002	0002	0	BCV002	0044 *	1467	SyncInProgress
DEV003	0003	0	BCV003	0045 *	7614	SyncInProgress
DEV004	0004	0	BCV004	0046 *	2305	SyncInProgress
DEV005	0005	0	BCV005	0047 *	18913	SyncInProgress
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 copy 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 multi-hop copy options defined in a file called `rep.txt` (for file content, refer to the `rep_opt.txt` file defined in Example 2).

```
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	Inv. Sym Tracks	Logical	Inv. Sym Tracks	STD <=> BCV	
DEV001	0001	0 BCV001	0043 *	0	Synchronized
		0 N/A	0031	0	Split
DEV002	0002	0 BCV002	0044 *	0	Synchronized
		0 N/A	0032	0	Split
DEV003	0003	0 BCV003	0045 *	0	Synchronized
		0 N/A	0033	0	Split
DEV004	0004	0 BCV004	0046 *	0	Synchronized
		0 N/A	0034	0	Split
DEV005	0005	0 BCV005	0047 *	0	Synchronized
		0 N/A	0035	0	Split
Total	-----			-----	
Track(s)	0			0	
MB(s)	0.0			0.0	

Legend:

(*): The paired BCV device is associated with this group.

Because the symreplicate copy cycle is still in progress, the following attempt 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'.

Example 7: Restarting a Replicate Session When Devices Are Locked

Device locks are held during the replicate session to block other applications from altering device states while this session executes. Under certain circumstances, a replicate session may exit with devices left in a locked state. For example, a replicate session may terminate when an RDF link goes down unexpectedly. Then the replicate session cannot restart after the RDF link is brought back up, because of the locked devices. Beginning with EMC Solutions Enabler version 5.2, you can use the `-recover` option with the `symreplicate start` or `restart` command to recover the existing device locks and restart the session (SRDF/AR checks if it previously owned the device locks and, if so, proceeds as if it just acquired the existing locks).

Using the `-recover` option allows you to recover without having to manually release the device locks. When SRDF/AR detects a situation where devices are locked and recovery is possible, SRDF/AR returns a message suggesting that you attempt to recover.

Caution: 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, `symdbg -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...
```