EMC® Solutions Enabler Symmetrix® TimeFinder® Family CLI

Version 7.0

Product Guide

P/N 300-000-876

REV A11

EMC Corporation

Corporate Headquarters:

Hopkinton, MA 01748-9103

1-508-435-1000

www.EMC.com

Published April, 2009

Contents

Chapter 10 Performing TimeFinder/Mirror Control Operations

Chapter 11 Performing Consistent Splits with TimeFinder Family

Appendix A SRDF State Rules Reference

Appendix B TimeFinder State Reference

Figures

Tables

Preface

As part of an effort to improve and enhance the performance and capabilities of its product lines, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

If a product does not function properly or does not function as described in this document, please contact your EMC representative.

Audience

This document is part of the Solutions Enabler documentation set, and is intended for use by advanced command-line users and script programmers to manage various types of control operations on Symmetrix arrays and devices using the SYMCLI commands of the EMC Solutions Enabler software.

Related documentation

Related documents include:

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

Note: Detailed man page descriptions of all SYMCLI commands, environment variables, option file parameters, and error codes are documented in the companion *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Conventions used in this document

In this document, every use of the word SYMCLI means Solutions Enabler. EMC uses the following conventions for notes, cautions, warnings, and danger notices.

Note: A note presents information that is important, but not hazard-related.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.

IMPORTANT

An important notice contains information essential to operation of the software. The important notice applies only to software.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal — Used in running (nonprocedural) text for:

- Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus)
- Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, utilities
- URLs, pathnames, filenames, directory names, computer names, filenames, links, groups, service keys, file systems, notifications

Bold — Used in running (nonprocedural) text for:

- Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, man pages

Used in procedures for:

- Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus)
- What user specifically selects, clicks, presses, or types

Italic — Used in all text (including procedures) for:

- Full titles of publications referenced in text
- Emphasis (for example a new term)
- Variables

`Courier` — Used for:

- System output, such as an error message or script
- URLs, complete paths, filenames, prompts, and command syntax

`Courier bold` — Used for specific user input (such as commands)

`Courier italic` — Used in procedures for:

- Variables on command line
- User input variables

< > — Angle brackets enclose parameter or variable values supplied by the user

[ ] — Square brackets enclose optional values

| — Vertical bar indicates alternate selections - the bar means “or”

{ } — Braces indicate content that you must specify (that is, x or y or z)

... — Ellipses indicate nonessential information omitted from the example

Where to get help

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

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

Technical support — For technical support, go to EMC Customer Service on Powerlink. To open a service request through Powerlink, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this document to techpubcomments@EMC.com.

PART 1

Concepts and Procedures

The Concepts and Procedures section of this product guide provides conceptual information and describes how to perform TimeFinder operations on Symmetrix devices using the SYMCLI commands of the EMC Solutions Enabler software. These concepts and procedures are described in the subsequent chapters as follows:

Chapter 1, "Overview," introduces the Solutions Enabler TimeFinder component and describes the types of devices used in TimeFinder configurations.

Chapter 2, "Performing TimeFinder/Clone Operations," describes how to perform TimeFinder/Clone operations using the SYMCLI `symclone` command.

Chapter 3, "Performing TimeFinder/Snap Operations," describes how to control copy sessions for virtual devices using the SYMCLI `symsnap` command.

Chapter 4, "Performing TimeFinder/Mirror Operations," describes the business continuance model and how to manage and control TimeFinder/Mirror (BCV) devices using SYMCLI. This is a legacy chapter relevant to Symmetrix environments running EMC Enginuity release levels 5773 and earlier. Starting with Enginuity release level 5874, the TimeFinder/Mirror functions are performed through TimeFinder/Clone software using a process called Clone Emulation.

Note: For practical examples illustrating how to perform specific array control tasks with SYMCLI command sequences, refer to *Part 2* of this guide.

1 Overview

This chapter introduces the EMC Solutions Enabler TimeFinder family components and their respective SYMCLI commands. It also describes the various Symmetrix device types that are used in TimeFinder configurations and explains common concepts pertinent to the devices and TimeFinder components.

Topics include:

Introduction to the EMC TimeFinder family

The Solutions Enabler EMC® TimeFinder® family of software consists of two base replication products and several component options. The TimeFinder family is used in environments configured with EMC Symmetrix® V-Max™ with Enginuity arrays and Symmetrix DMX™ arrays. Symmetrix V-Max arrays require EMC Enginuity™ Operating Environment for Symmetrix systems release 5874 or later. Symmetrix DMX arrays require Enginuity 5773 and earlier.

Note: Features that are new in Enginuity 5874 are not supported on arrays running Enginuity 5773 and earlier unless noted.

- Base replication products:
  - **TimeFinder/Clone** provides clone copy sessions that create point-in-time copies of full volumes or individual datasets.
  - **TimeFinder/Snap** provides snap copy sessions that create economical, pointer-based replicas where only the pre-images of changed data are written.
- Component options:
  - **TimeFinder/Mirror** is another family component that works in Symmetrix DMX environments running Enginuity 5773 and earlier. In environments running Enginuity 5874 and higher, all TimeFinder/Mirror scripts are executed in Clone Emulation mode.

IMPORTANT

Starting with Enginuity 5874, TimeFinder/Mirror functions are performed through TimeFinder/Clone software using a process called Clone Emulation. When running in Emulation Mode, TimeFinder/Clone transparently performs TimeFinder/Mirror commands and executes scripts that were written for Solutions Enabler up through version 6.5.2 running on Symmetrix arrays using Enginuity 5773 and earlier. **"TimeFinder/Clone Emulation" on page 120** **provides greater detail.**

  - **TimeFinder/Consistency Groups** (**TimeFinder/CG**) enables other TimeFinder family products to coordinate cross-volume and cross-system consistency to ensure application restartability. This option for Symmetrix arrays creates multivolume sets of point-in-time copies at the same instant, ensuring that as a set the copies are consistent and restartable, even when spread across multiple Symmetrix volumes and data is spread across multiple arrays, without quiescing or shutting down to ensure that all devices are copied at the same point in time.
  - **TimeFinder/Exchange Integration Module** (**TimeFinder/EIM**) automates and simplifies the process of creating and managing TimeFinder replications of a Microsoft Windows Exchange Server environment.
  - **TimeFinder/SQL Integration Module** (**TimeFinder/SIM**) automates (TimeFinder/SIM) simplifies the process of creating and managing TimeFinder replications of a Microsoft Windows SQL Server environment.

Note: The TimeFinder components complement one another to ensure maximum replication coverage for the entire data processing environment. Of the component options, only TimeFinder/Mirror is completely documented in Chapter 4, "Performing TimeFinder/Mirror Operations."

Configuring and controlling remote device pairs requires SRDF® business continuity software. The combination of TimeFinder with SRDF provides for multiple local andremote copies of production data. For greater detail on SRDF and the SRDF family of products, refer to the EMC Solutions Enabler SRDF CLI Product Guide.

Note: Each TimeFinder component described herein requires a separate license, unless noted otherwise.

The SYMCLI TimeFinder component extends the basic SYMCLI command set to include TimeFinder or business continuance commands that perform control operations on device pairs within the TimeFinder environment. The commands that comprise the TimeFinder component technologies of the EMC Solutions Enabler are: `symclone`, `symsnap`, `symbcv`, `symmir`, and `symioctl`. Table 1 on page 25 summarizes the TimeFinder control operations that these commands perform.

- The TimeFinder/Clone `symclone` command creates a point-in-time copy 2h as standard/standard and BCV/BCV).

  The TimeFinder/Clone operations are `create`, `activate`, `recreate` and `activate` (or `establish` which combines the two procedures in one operation), `restore`, and `terminate`. Chapter 2 describes the `symclone` command and the TimeFinder/Clone operations in great detail.

- The TimeFinder/Snap `symsnap` command creates virtual device copy sessions between a source device and multiple *virtual* (VDEV) target devices. These virtual devices only store pointers to changed data blocks from the source device, rather than a full copy of the data.

  The TimeFinder/Snap operations are `create`, `activate`, `recreate`, `restore` and `terminate`. Chapter 3 describes the `symsnap` command and the TimeFinder/Snap operations in great detail.

- Base component commands such as `symmir` and `symbcv` perform a wide spectrum of monitor and control operations on Symmetrix standard/BCV device pairs within a TimeFinder environment. The symioctl command sends I/O control commands to a specified database server.

To ensure a proper understanding of the control actions that the TimeFinder SYMCLI commands perform on device pairs:

- A brief description (beginning on page 20) of the device types discussed throughout this guide is provided.
- Table 1 on page 25 summarizes the control actions that the TimeFinder SYMCLI commands perform in their respective sections in Chapters 2 through 4.

Symmetrix device types used in TimeFinder operations

This section introduces the Symmetrix device types (standard, BCV, Clone, virtual, SAVE, and so on) that are used in the TimeFinder configurations discussed throughout this guide.

Standard device

A standard device is a Symmetrix storage device (disk) that is configured for normal Symmetrix operations under a desired protection method (SRDF, RAID S, RAID 1, RAID 5, RAID 6, and so on).

Standard devices can have any mirror structure (unprotected, one mirror, two mirrors), provided the number of mirrors does not exceed three. This constraint is in place because establishing a "BCV device" pair requires assigning the BCV device as the next available mirror of the standard device. When TimeFinder/Clone is in emulation mode, this constraint does not apply — the BCV device is not assigned as the next mirror.

Note: Devices that are *metahead* devices can be standard devices. *Metamembers* are implicitly controlled when the metahead is controlled (refer to "Metadevice" on page 22).

BCV device

A BCV device, or Business Continuance Volume, is a standard Symmetrix device that contains a copy of data from a standard Symmetrix device that is online for regular I/O operation from one or more of its hosts. The BCV is a full volume mirror or spare that has valid data after fully synchronizing with its paired source device. The BCV is accessible only when split from the source device that it is mirroring.

Uses for the BCV copies can include backup, restore, decision support, and applications testing. Each BCV device has its own host address, and is configured as a stand-alone Symmetrix device. with special attributes that allow it to be accessed independently of its associated standard device to support host applications and processes:

- BCV devices can also be metahead devices (see page 22) and the metamembers will be implicitly controlled when the BCV metahead is manipulated.
- BCV devices cannot be thin devices (see page 23). A full volume mirror has valid data after fully synchronizing with its source device and is accessible only when split from the source device that it is mirroring.

A Business Continuance sequence first involves *establishing* the BCV device as a mirror of a specific standard Symmetrix device. As a result, the BCV device becomes inaccessible via its original device address while it is in an established pair. Once the BCV device is synchronized, it may be separated (*split*) later from the standard device with which it is paired. Once split, the BCV device with the synchronized data becomes available for backup or other host processes through its original device address. Once host processing on the BCV device is complete, the BCV may again be mirrored to a standard Symmetrix device, which can either be the same device to which it was previously paired, or a different device.

Clone device

A clone device is a full-volume Symmetrix device used in TimeFinder/Clone operations to create a physical point-in-time Copy of a logical standard device on multiple target devices. The data is physically copied from the standard device, creating a physical backup copy, which can be used in case the STD becomes inaccessible.

TimeFinder/Clone copies are appropriate in situations where multiple copies of production data are needed for testing, backups, or report generation. Clone copies can also be used to reduce disk contention and improve data access speed by assigning users to copies of data rather than accessing the one production copy.

The source and target devices can be either standard devices or BCV devices as long as they are all of the same size and emulation type (FBA or CKD). Clone copies of striped or concatenated metadevices can also be created, but the source and target metadevices must be completely identical in stripe count, stripe size, and capacity. Once activated, the copy can be instantly accessed by a target's host, even before the data is fully copied to the target device.

Virtual device (VDEV)

A virtual device (VDEV) is a Symmetrix host-accessible, logical-image device that contains track-level location information (pointers) indicating where the copy session data is located in the physical storage and is immediately accessible after activation.

Virtual devices are used in TimeFinder/Snap operations and consume minimal physical disk storage and contain only the address pointers to the data that is stored on the source device or in a pool of *"SAVE devices."*

When a virtual copy session is activated, a point-in-time copy of the source device is immediately available to its host through the corresponding virtual device. Snapping data to a virtual device uses a copy-on-first-write technique. Upon a first write to the source device during the copy session, a preupdated image of the changed track is copied to a SAVE device. The track pointer on the virtual device will then be updated to point to the data on the SAVE device.

The attached host views the point-in-time copy through virtual device pointers to both the source device and SAVE device, for as long as the session remains active. If the copy session is terminated, the copy is lost and the space associated with the session is freed and returned to the SAVE device pool for future use. Optionally, before terminating a copy session, virtual devices can be incrementally or fully restored back to the original source, a BCV device split from its source, or another Symmetrix device (refer to "Restoring data from virtual devices" on page 82).

Note: Virtual and SAVE devices must be configured on the Symmetrix array by using the Symmetrix Configuration Manager software. The *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* discusses how to configure devices using the SYMCLI.

SAVE device

A SAVE device, or SAVEDEV, is a Symmetrix device that is not host-accessible and can only be accessed through the virtual devices that point to it. SAVE devices are configured to provide pooled physical storage to TimeFinder/Snap and SRDF/A operations:

- With TimeFinder/Snap, SAVE devices are organized into snap pools and are used to store pre-update images or changed tracks during a virtual copy session. Devices within a snap pool act as a group for storing data in striped form. Symmetrix supports the creation of multiple named snap pools, allowing `symsnap` commands to use a particular pool.

When used in snap pools, SAVE devices contain any original tracks that were changed as a result of a first *copy on write* to a source device or a new write to a virtual device during a virtual device copy session. "Activating a virtual copy session" on page 80 provides more information on how SAVE devices are implemented in a copy session.

- With SRDF/A, SAVE devices are organized into *Delta Set Extension* pools and are used to extend the cache space available for SRDF/A cycles by offloading some or all of the cycle data from cache to preconfigured disk storage pools. For more information on RDF/A DSE pools, refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide*.

SAVE devices are assigned a Symmetrix device number and can be configured as unprotected, mirrored, or parity RAID. They cannot be part of a metavolume or grouped in device or composite groups.

Note: EMC does not recommend configuring SAVE devices as unprotected; doing so requires special approval by EMC.

SAVE devices are configured with a limited amount of available storage space and require careful planning before use. SAVE devices can be monitored for the percentage full and can be expanded.

Metadevice

A metadevice, or metavolume, consists of two or more individual devices logically concatenated, creating larger storage devices and presented to the host as a single addressable device. The Metahead is the first device in the chain of metadevices or metamembers and is the initial target as it is a responsible gatekeeper for keeping incoming commands. The metahead handles all command processing activities. When the Metahead is addressed with a command, Enginuity determines which metadevice is the target for the command execution.

Besides concatenation, metavolumes can be striped for better performance. A striped metadevice is one that places data on metamembers in user-defined stripes or chunks instead of filling an entire device first before addressing the next device. The striping of data across multiple drives effectively creates definable cylinder stripes.

Note: Beginning with Solutions Enabler V6.5.1 running Enginuity 5773, metadevices can automatically be created using a single configuration change session. The *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* provides complete details on metadevices.

DATA device

A DATA device is a specifically configured devices within the Symmetrix array that is not addressable to a host and is a container for the written-to blocks of thin devices. Any number of DATA devices may comprise a data device pool (also referred to as a thin pool). Blocks are allocated to the thin devices from the pool on a round robin basis. This allocation block size is 768 KB.

When DATA devices are added to a thin pool, the devices can be in an enabled or disabled state. To use a DATA device for thin extent allocation, the devices must be in the enabled state. To remove devices from the thin pool, the devices must be in a disabled state. The *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* provides greater detail on how DATA devices are configured.

Thin device

A thin device is a host-accessible, logical Symmetrix device used in ways similar to the ways Symmetrix devices have traditionally been used. Like traditional devices, thin devices are provisioned by mapping them to a front-end Symmetrix director and port, and then LUN-masking them to the WWN of an HBA in the server requiring the storage.

Unlike traditional Symmetrix devices, thin devices have no storage directly associated with it — they do not require physical storage completely allocated at the time the device is created and presented to a host. They have preconfigured sizes and appear to the host to have that exact capacity. Device storage is allocated in chunks when a block is written to for the first time. Zeroes are provided to the host for data that is read from chunks that have not yet been allocated. The physical storage that is used to supply disk space (storage capacity) to thin devices comes from a shared storage pool called a thin pool.

Concurrent SRDF devices

Concurrent RDF operations use the following Symmetrix device types:

- **R11 device** — Available with Enginuity 5773 and later, an R11 device (also known as a concurrent R1) is a source device that allows data to be replicated to two R2 devices in different Symmetrix arrays.
- **R21 device** — Available with Enginuity 5773 and later, an R21 device configured for Cascaded SRDF operations, can be designated as either a source device (R1), a target mirror (R2 device), or a cascaded mirror (R21) device. In other words, in Cascaded SRDF operations, the R21 assumes the dual roles of primary (R1) and secondary (R2) simultaneously.

  A basic Cascaded SRDF configuration consists of a primary or workload site (site A) replicating to a secondary site (site B) and then replicating the same data to a tertiary site (site C). If the source device fails, the data on its corresponding target device can be accessed by the local host. Once the source device is replaced, it can be resynchronized. For example, the R21 device sits at the synchronous target site mirroring the R1 device at workload production site, while at the same time, behaves like an R1 device in relationship to the third (hop 2) site or asynchronous target site's R2 mirroring device. The workload site A and secondary site B have an SRDF pair state; secondary site B and tertiary site C have and SRDF pair state. These two pair states are separate from each other, but each must be considered when performing a control operation on the other pair.
- **R21 Diskless Device** — Available with Enginuity 5874 and later, an I/O connected R21 Diskless Device (DL Dev) is a Symmetrix device that does not have any local mirrors. Since the device has no local disk space allocated to store user data, this reduces the cost of disk storage in the Symmetrix array by directly cascading data from the R21 DL Dev to the remote R2 device, streamlining the linkage. With R21 diskless devices, the changed tracks received from its R1 mirror are saved in cache until the tracks are sent to the R2 device. Once the data is sent to the R2 device and the receipt is acknowledged, the cache slot is freed and the data no longer exists on the R21 device.

  Note: R21 Diskless Devices are also used in SRDF/Extended Distance Protection with Star environments and are supported in conjunction with the SRDF/A Delta Set Extension feature.

- **R22 devices** — Available with Enginuity 5874 and Solutions Enabler v7.0, an R22 device (also referred to as a concurrent R2) is a preconfigured device that acts as the target device to two source R1 devices. In multisite environments, using R22 devices simplifies failover (swap) operations by automating the decision making as to what devices to pair up if a failure occurs. R22 devices are most commonly used in configurations consisting of two R2 mirrors or an R22 device paired with the R1 mirror of an R11 or R21 device. Only one of the R2 mirrors can be Read/Write on the link at a time.

  There are many advantages of using R22 devices. For example, when a source device relationship needs to change using R22 devices eliminates the need for Delete and Create Pair operations. This operation differs from earlier R2 device operations where performing pair operations were required when creating new relationships to a new R1 device. Given the ease of changing device relationships, R22 devices are ideal for Cascaded Star environments.

  Note: An R22 device can only actively accept I/Os from one of the source device at a time. An R22 device cannot be a BCV device — Enginuity 5874 does not support more than one RDF mirror.

Command summary

Note: To use the SYMCLI TimeFinder commands described herein, make sure the SYMCLI environment is set up as described in the *EMC Solutions Enabler Installation Guide*.

As previously mentioned, the `symbcv`, `symmir`, `symclone`, `symsnap`, and `symioctl` commands make up the TimeFinder component technologies of the EMC Solutions Enabler. These commands perform control operations on device pairs, TimeFinder device groups, or TimeFinder composite groups. Table 1 lists and summarizes these commands. The *EMC Solutions Enabler Symmetrix CLI Command Reference* provides complete detail about the command syntactical form.

Table 1 **TimeFinder command summary (page 1 of 2)**

Command	Description
`symclone`	Performs TimeFinder/Clone control operations on standard or BCV devices: • Creates a copy session for making multiple data copies between a source device and up to 16 target devices (15 clone copy sessions and one session reserved for restore operations). • Creates and activates a copy session • Modifies the mode in which a copy session is operating. • Activates a copy session to make data instantly accessible to multiple target hosts. • Copies (incrementally) all subsequent changes made to a source device to a target device, after a clone session is fully copied. • Restores data from a target device back to a source device or to another device. • Terminates a copy session to remove holds on target devices and delete device pair information from the Symmetrix array. • Queries information about the state of mirroring for multiple copy sessions. • Verifies the state for selected devices. • Lists all copy sessions that have been created on the Symmetrix array.
`symsnap`	Performs Snap control operations for virtual copy sessions from normal Symmetrix devices to virtual devices. The source device can be either a standard or a BCV device and the target device must be a virtual device (VDEV). • Creates a virtual copy session for making multiple data copies between a source device and up to 15 target devices. The number of target devices increases to 128 when using multivirtual snap. • Specifies a particular SAVE device pool for use in a virtual copy session. • Activates a virtual copy session to make data instantly accessible to multiple target hosts. • Recreates a snap session on existing VDEVs to prepare to active a new point-in-time image and is only valid when issued against previously activated sessions. • Terminates a virtual copy session to remove holds on target devices and delete device pairing information from the Symmetrix array. • Queries information about the state of mirroring for multiple copy sessions. • Verifies device states. • Attaches and detaches target devices as the preferred devices to use in a requested Snap operation. • Restores a virtual device to another device, or to the original device. • Monitors the total percentage full of SAVE devices in a virtual copy session. • Lists all virtual copy sessions that have been created on the Symmetrix array.

Table 1 TimeFinder command summary (page 2 of 2)

Command	Description
`symbcv`	Performs operations on one or more Symmetrix BCV devices: • Associates a device pair. • Disassociates a device pair. • Lists all BCV devices in the Symmetrix array. • Moves a BCV device from one group to another. • Removes all BCV devices from the specified device group.
`symmir`	Performs control operations on BCV device pairs including: • Establishes (mirror) one or all standard devices with one or more BCV devices. The operation can be a full or incremental establish. • Restores one or all standard devices from one or more BCV devices that are associated locally or remotely. The operation can be a full or incremental restore. • Splits one or all BCV devices from one or more standard devices. • Returns information about the state of mirroring of one or all BCV device pairs. • Cancels the existing internal SDDF session between the specified standard and BCV devices. • Lists all BCV sessions created on a Symmetrix array.
`symioctl`	Sends I/O control commands to a specified server application. • Places objects into hot backup mode. • Freezes or thaws I/O to a specified database application. • Issues a checkpoint to the RDBMS. • Archives the current log. • Begins a Snap backup on SQL Server 2000 and higher. • Saves Snap metadata and resumes writes on SQL Server 2000 and higher. • Restores previously saved Snap metadata on SQL Server 2000 and higher. • Terminates the Snap operation without saving metadata and resumes writes on SQL Server 2000 and higher.

Common TimeFinder command operations using SYMCLI

To help ensure a basic understanding of concepts discussed in the TimeFinder family component chapters in Part 1 and their operational examples in Part 2, the sections that follow provide a glimpse into functionality that is common to the TimeFinder family — the difference being the command and the action performed on the device pair type.

Command scope

The scope of a TimeFinder, Clone, and Snap command determines which devices are to be considered for the operation. This scope is defined as the set of devices in a device group (DG) or composite group (CG) or the set of devices explicitly specified using the DG or CG LDEV syntax. Currently, if the source devices in the DG or CG have any TimeFinder/Clone or TimeFinder/Snap sessions with devices outside the DG or CG, the operation will be performed on the source devices contained in the DG or CG, while considering those sessions whose target devices are outside of the DG or CG when determining if the source device is already in the desired state.

Command Scope limits the device selection process to only consider the source devices and sessions with target devices contained within the DG or CG when considering if the source device is already in the desired state.

Solutions Enabler V7.0 has an environment variable, SYMCLI_COMMAND_SCOPE, that enables the SYMCLI to override the Command Mode feature default set in the options file with the SYMAPI_COMMAND_SCOPE option.

When limiting control commands with `COMMAND SCOPE` mode enabled, the source device selection criteria only considers pairings with potential targets that are within the scope of the command either contained in the DG or CG for group commands or explicitly contained within a device list or logical device list. This means that return codes such as `ALREADY IN BCV STATE` will not occur as the result of a session whose target is outside of the scope of the devices that are the target of the command.

Group operations

Starting with Solutions Enabler Version 5.4, TimeFinder operations using the SYMCLI `symmir`, `symclone`, and `symsnap` commands support composite groups (`-cg`) or devices in a composite group, as well as device groups (`-g`) and devices within a device group. These operations are discussed in their respective TimeFinder component chapters:

Composite groups:

- For TimeFinder/Clone environments — Chapter 2 “Using composite groups to manage clone pairs across Symmetrix arrays” on page 65
- For TimeFinder/Snap environments — Chapter 3 “Using composite groups to manage snap pairs across Symmetrix arrays” on page 96
- For TimeFinder/Mirror environments running Enginuity release level 5773 and earlier — Chapter 4 “Using composite groups to manage BCV pairs across Symmetrix arrays” on page 196

Device groups used with composite groups:

The following tables list the TimeFinder control operations and the possible operations to use when targeting a specific device group used with composite groups:

- Table 5, “symclone -g control arguments and possible options,” on page 67
- Table 6, “symclone -cg control arguments and possible options,” on page 69
- Table 8, “symsnap -g and -cg control arguments and possible options,” on page 108

- Table 19, "symmir -g control arguments and possible options," on page 213
- Table 23, "symmir -cg control arguments and possible options," on page 222

Note: In addition to TimeFinder composite group operations listed in the tables above, the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* contains complete detail on composite groups and device groups.

Command options with device pairs

The following tables list the TimeFinder control operations and the possible options to use when targeting device pairs specified in a device file of a given Symmetrix array:

- Table 7, "symclone -file control arguments and possible options," on page 71
- Table 9, "symsnap -file control arguments and possible options," on page 110
- Table 25, "symmir -file control arguments and possible options," on page 225

Cascaded SRDF operations

Starting with Solutions Enabler Version 6.5, the SYMCLI TimeFinder commands `symclone`, `symsnap`, and `symmir` can query and manage devices in a Cascaded SRDF configuration.

Note: The *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide* provides greater detail on Cascaded SRDF.

2

Performing TimeFinder/Clone Operations

This chapter describes how to perform TimeFinder/Clone operations using the SYMCLI `symclone` command.

TimeFinder/Clone environment

Symmetrix TimeFinder/Clone operations are performed using the SYMCLI TimeFinder `symclone` command which creates *clone* copies of a source device on multiple target devices. The source and target devices can be either standard devices or BCV devices as long as they are all of the same size and emulation type (FBA or CKD). Clone copies of striped or concatenated metadevices can also be created, but the source and target metadevices must be completely identical in stripe count, stripe size, and capacity. Once activated, the copy can be instantly accessed by a target's host, even before the data is fully copied to the target device.

There are several key advantages of using TimeFinder/Clone, such as the ability to perform precopy operations and its cache partitioning. TimeFinder/Clone copies are appropriate in situations where multiple copies of production data are needed for testing, backups, or report generation. Clone copies can also be used to reduce disk contention and improve data access speed by assigning users to copies of data rather than accessing the one production copy.

Depending on whether a device has associated BCVs, a single source device can have up to 16 clone copy sessions (15 copy sessions and one reserve copy session for restore operations). When using the `-copy` option, you can copy up to eight full data copies simultaneously, without disruption to database production activity.

Note: With Enginuity versions prior to 5671, you can only copy up to four full data copies simultaneously. "Understanding copy session limits" on page 32 contains greater detail on the number of available copies.

Clone operations overview

TimeFinder/Clone functionality is controlled through copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix array and can be queried to verify the current state of the device pairs. A copy session must first be created to define and set up the clone devices. The session is then activated, enabling the target device to be accessed by its host. When the information is no longer needed, the session can be terminated.

TimeFinder/Clone operations are controlled from the host by using the `symclone` command to `create`, `activate`, `restore`, `recreate`, `set mode`, `split`, `establish`, and `terminate` the clone sessions. These operations are discussed in great detail in the sections that follow. This chapter explains how to manage the devices participating in a copy session using SYMCLI. Figure 1 illustrates a clone session where the controlling host creates a clone copy of standard device `DEV001` on target device `DEV005`. "Creating a clone copy session" on page 32 discusses this operation.

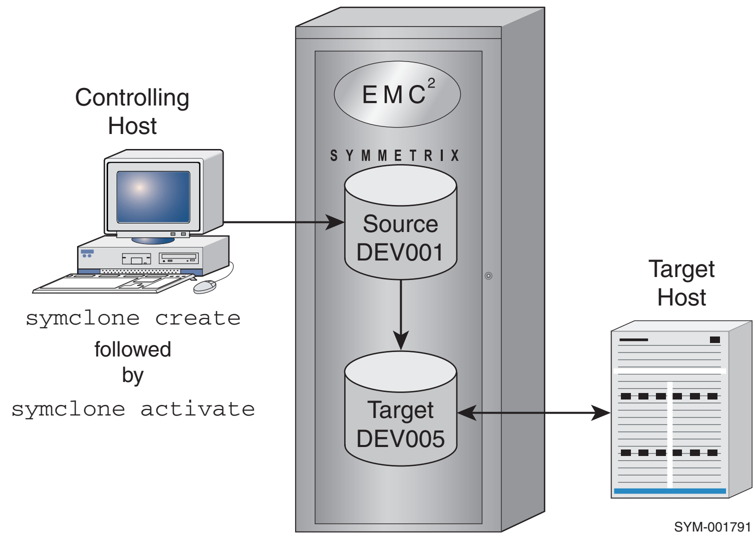

Figure 1 **TimeFinder/Clone copy of a standard device**

Creating a clone copy session

Initially, you must create a clone copy session that defines and sets up the cloning devices you have selected for the clone operation. For example, to begin a clone copy session and define a specified target device `DEV005` to be the clone of source device `DEV001` in group `ProdDB`, enter:

```
symclone -g ProdDB create DEV001 sym ld DEV005
```

The `symclone create` action defines the clone copy session requirements and sets the track protection bitmap on the source device to detect which tracks are being accessed by the target host or written to by the source host:

- The target device is made Not Ready to its host and placed on hold status for clone copy session activity. This prevents other control operations from using the device.
- The device pair state will transition from CreateInProgress to Created when complete.
- The clone copy does not become accessible to its host until the copy session is activated. "Activating a clone copy session" on page 39 contains greater detail.

Note: You cannot verify that a device pair is in the CreateInProgress state. Once the copy session completes, you can issue a `symclone verify -created` command to verify that the clone pair was successfully created. If a copy session is created and not activated, it can be terminated. Although, the data on the target device should then be considered invalid.

Understanding copy session limits

The Symmetrix array is currently limited to 16 sessions per source device, which can be used for TimeFinder/Clone, TimeFinder/Snap, SRDF/Star, Solutions Enabler Open Replicator (ORS), or Symmetrix Differential Data Facility (SDDF) operations. This limits the number of available clone copies that can be created. For example:

- When a standard device has two BCV device pairs, you can only create up to 14 copy sessions for that standard device.
- When TimeFinder/Clone is running in emulation mode, this number is decremented an additional two sessions per single BCV relationship.
- Copy sessions created using the `-copy` option to create full data copies are limited to eight concurrent sessions, with up to an additional eight CopyOnAccess sessions, for a total of 16 sessions.
- Copy sessions created with the `-differential` option uses two session slots.

Table 2 on page 33 lists the number of copy sessions required for each operation:

Table 2 **Number of session slots used per operation**

Operation	Session slots
TimeFinder/Snap	One session slot per snap session, plus an additional session reserved for restore operations.
Multivirtual snap	One session slot for all multivirtual snap sessions, plus an additional session slot reserved for restore operations.
TimeFinder/Clone	One session slot per copy session, unless using the `-differential` option, in which case two session slots are used per copy session.
TimeFinder/Clone Emulation mode	Two session slots.
ORS	One session slot per ORS session, and one session slot for each additional session created with the `-differential` option.
SRDF/Star	Two session slots.
SRDF/A	One session slot.
SDDF	One session slot per SDDF session

Creating in Nocopy mode

When activating a copy session, the default device pair state is *CopyOnAccess*. This means that after activating the copy session, only those tracks that have been written to the source or written/read from the target will be copied to the target device. A full data copy to the target device will *not* occur unless all of the device tracks are accessed or written to while participating in the active session.

Note: If a write occurs to the source device, old data is copied to the target device. If a write occurs to the target device, new data is written to the target device.

When Enginuity detects that a source-protected track was written, it copies the track to the target device and unprotects the track before accepting the new write. Data from the source then becomes available to a target-connected host during the active session.

You can modify this default device pair state to *CopyOnWrite* by setting the following parameter in the `options` file to `ENABLE`.

```
SYMAPI_CLONE_COPY_ON_WRITE = ENABLE | DISABLE
```

Once you have enabled *CopyOnWrite* as the default pair state and activated a copy session, all reads will be handled from the source device and writes to the source device or target device during the active copy session will result in the data being copied to a target device.

Fully copying to a clone using the -copy option

The following example uses the `create` argument with the `-copy` option that defines a full copy clone action to be run in the background:

```
symclone -g ProdDB create DEV001 sym ld DEV005 -copy
```

When the copy session is eventually activated, data begins background copying so that a full copy of the data will become available on the target device. While background copying, the state of the device pair is Copy In Progress; when the operation completes, the state goes to Copied. The copy session must be activated before the target host can access the data. However, once the session is activated, the data is available to the target host immediately.

By omitting the `-copy` option (default), the device pair state will be in the CopyOnWrite/CopyOnAccess state when activated. Actual copying of the data is deferred until either tracks on the source device are written to or tracks on the target device are read or written. "Creating in Nocopy mode" on page 33 contains greater detail.

Starting to copy data before activating the session

With Enginuity Version 5671 and later, you can use the `-precopy` option with the `create` argument to start copying tracks in the background, before activating the copy session. This allows the early movement of data before the point-in-time clone copy is established.

Note: When using this option, the target device is not ready to the host until the session is activated.

Pre-copying can occur in the Created or Recreated state. While in this state, the pre-copy process keeps checking for new writes to be pre-copied to the target device until the copy session is activated. Once activated, the normal background copy mechanism starts and the pre-copy operation ends. You can set the pre-copy mode in one of two ways:

- Include the `-precopy` option with the `symclone create` command. To set up incremental copying on subsequent copy operations, you need to *add* the `-differential` option (described later on this page). Using `-precopy` will result in a full copy. For example:

  ```
  symclone -g ProdDB create -precopy SRC sym ld TGT -differential
  ```

- Use the `symclone set mode` command as described in "Modifying a clone copy session" on page 42. For example:

  ```
  symclone -g ProdDB set mode precopy SRC sym ld TGT
  ```

Copying only changed data to a clone

With Enginuity Version 5671 and later, subsequent cloning to the same target can be performed as differential copying. You can use the `-differential` option the first time you create a clone copy session. All subsequent copying during that copy session automatically becomes incremental (that is, copying only new writes to the source device).

Note: EMC does not support the `-differential` option in Nocopy mode.

A differential or incremental clone operation copies only those device tracks that have changed since the initial full clone was performed. To do this, however, the copy session that existed for the full clone must still exist. You must also set up the differential clone during a full clone operation. The `-differential` option, therefore, must be used here with the `-copy` (or `-precopy`) option. For example:

```
symclone -g ProdDB create -copy -differential SRC sym ld TGT
```

The `–differential` option creates an SDDF session for the source and target.

The `symclone activate` command initiates a full copy of the source device to the target device. For example:

```
symclone -g ProdDB activate SRC sym ld TGT
```

To capture subsequent writes to the source during this clone pair session, use `symclone recreate` to clone just those tracks that have changed since the full copy completed. For example:

```
symclone -g ProdDB recreate SRC sym ld TGT
symclone -g ProdDB activate SRC sym ld TGT
```

To save some steps, you can use the `symclone establish` command in the same way. For example, after the first full clone operation, you can capture any new writes with a subsequent clone operation that is automatically an incremental copy:

```
symclone -g ProdDB establish -full SRC sym ld TGT
:
:
symclone -g ProdDB establish SRC sym ld TGT
```

Copying from a source device to a larger target device

Clone source and target devices no longer have the requirement to be the same size. Now the size of a clone target device can be larger than the source device. This support requires the following SYMCLI environment variable be set:

```
SYMCLI_CLONE_LARGER_TGT = ENABLED
```

Limitations — The following limitations apply:

- Restore is not allowed.
- Full copy support only; cannot use `-differential`.
- Concatenated metadevices are not supported.
- When using this feature on striped metadevices you have to preserve the metageometry. That is, the source and target devices should contain the same number of metamembers. However, the target device members can be larger than the source device members.
- Exact pairing are the only operations allowed, as follows:

  `-file`

  `-g` or `-cg` with `-exact`

  `-g` or `-cg` with source and target ldev name supplied

Comparing copy functionality between TimeFinder/Clone and TimeFinder/Snap

Table 3 compares copy functionality between TimeFinder/Clone operations for standard and BCV devices with TimeFinder/Snap operations for virtual devices.

Table 3 Comparing target type

Copy functionality	Target is a standard or BCV	Target is a virtual device
Minimum Enginuity Level	5568 [a]	5669 [b]
Copy on first read to target	Yes [c]	No
Copy on first write to source	Yes	Yes
Copy on first write to target	Yes	Yes
Use of the -copy option	Yes	No
Incremental restore	Yes [d]	Yes
Full restore	Yes	Yes
Precopy	Yes [d]	No
Recreate	Yes [d]	Yes[e]

a. Full Device and CopyOnAccess TF Clone copies for open systems using the `symclone` command.

b. Virtual TF/Snap Copy and Restore for open systems using the `symsnap` command.

c. For Enginuity Version 5772, you can use CopyOnWrite.

d. Supported with Enginuity Version 5671 and later.

e. Supported with Enginuity Version 5874 and later.

Using the establish command

To create and then immediately activate a copy session with a single command, you can use the `symclone establish` command.

For example, to create and then activate the copy session shown in the example on page 34, enter:

```
symclone -g ProdDB establish DEV001 sym ld DEV005 -full
```

This example implies the `-copy` and `-diff` options.

Note: The `symclone establish` command sets the target device to Not Ready for a short time. Therefore, you may want to unmount the target host before issuing the command.

Pairing an additional target device with each source device in a group

When working with either a composite or device group, you can use the `-concurrent` option with the `create` or full `establish` action to pair an additional target device with each source device in a group.

For example, to pair an additional target device with each source device in group `ProdDB`, enter:

```
symclone -g ProdDB create -concurrent
```

When the copy session is created, an additional target device will be paired with each source device in the group. For example, if there were two target devices paired with each source device in the group before creating the session, there will be three target devices paired with each source device after the session is created.

To verify that each source device in the group has multiple targets, enter:

```
symclone -g ProdDB verify -created -concurrent
```

Performing operations on devices in a clone target list

When working with either a composite or device group, you can use the `-tgt` option to indicate that devices from a *local* target list are to be used as targets for the specified action. You can also use the `-tgt` option together with the following options to indicate that devices from a *remote* target list are to be used as targets for the specified action:

- `-rdf` specifies remote attached BCVs (RBCVs).
- `-bcv` specifies remote BCVs that mirror local BCVs (BRBCVs). This option must be used with the `-rdf` option.
- `-rrdf` specifies BCVs that are remotely associated with remote BCVs (RRBCVs).
- `-hop2` specifies BCV devices (2BCVs) that are remotely associated on the second hop of a Cascaded SRDF configuration.

The `-tgt` option will work with all symclone actions, except for `query` and `verify`, as these actions are source device oriented and will select all target devices paired with source devices by default, including TGT and RTGT.

When working with specific pairs, the following symclone syntax will support the target devices:

```
sym ld LdevName
sym dev SymDevName
sym pd PdevName
```

Note: The *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide contains* information on creating and managing clone target lists.

In the following examples, *`action`* can be any symclone actions, except for `query` and `verify`.

Copying from a local STD to a local TGT

To copy from a local standard device to a local target, use the following syntax:

For device groups:

```
symclone -g DgName action -tgt
```

For composite groups:

```
symclone -cg CgName action -tgt
```

Copying from a local BCV to a local TGT

To copy from a local BCV device to a local target, use the following syntax:

For device groups:

```
symclone -g DgName action -tgt -bcv
```

For composite groups:

```
symclone -cg CgName action -tgt -bcv
```

Copying from a STD’s remote partner to a remote RTGT

To copy from a STD’s remote partner to a remote RTGT, use the following syntax:

For device groups:

```
symclone -g DgName action -tgt -rdf
```

For composite groups:

```
symclone -cg CgName action -tgt -rdf
```

Copying from a RBCV to a remote RTGT

To copy from a remote BCV to a remote RTGT, using the following syntax:

For device groups:

```
symclone -g DgName action -tgt -rbcv
```

For composite groups:

```
symclone -cg CgName action -tgt -rbcv
```

Copying from a hop 2 BCV to a remote RTGT in a Cascaded SRDF configuration

To copy from a second hop BCV to a remote RTGT, use the following syntax:

For device groups:

```
symclone -g DgName action -tgt -hop2
```

For composite groups:

```
symclone -g CgName action -tgt -hop2
```

Activating a clone copy session

To activate the copy session created in the example in "Fully copying to a clone using the -copy option" on page 34, enter:

```
symclone -g ProdDB activate DEV001 sym ld DEV005
```

This activates the copy operation from the source device to the target device. Activating the copy session places the target device in the Read/Write state and initiates the `-copy` option if it was specified when the session was created. The target host can access the cloned data and has access to data on the source host until the copy session is terminated.

Note: Cloned data is made available as a point-in-time copy at the time of activation and not at the time that the session was created.

Precopying data before activating the session

The `-precopy` option can be used with the `create` or `recreate` actions to start copying tracks in the background, before the copy session is activated. When using this option, a point-in-time copy will be established when the session is activated.

While in the Created state, the pre-copy process never actually completes. Instead, the process keeps checking for new writes to be pre-copied to the target device until the session is activated. Once activated, the normal background copy mechanism finishes copying the remaining tracks and the pre-copy operation ceases.

Making the target device not ready to the host

The `-not_ready` option can be used with the `activate` action to cause the target device to remain not ready to its host, as follows:

```
symclone -g ProdDB activate DEV001 sym ld DEV005 -not_ready
```

The copy session will be activated and the target device will be placed in the Not Ready state. The clone copy can later be read/write enabled to the host using either the `symld ready` or `symdev ready` command.

Activating copy sessions consistently

The `symclone activate` command can be used with the `-consistent` option to create clone copies that are consistent with the database up to the point in time that the activation occurs. This feature can be implemented using either PowerPath-connected devices, the Consistency Assist feature,[1] or SRDF/A.

This feature can be used to create clone copies that are consistent with the database up to the point in time that the activation occurs. The feature suspends writes to the source devices during the activation.

When the activation has completed, writes are resumed and the target device contains a consistent production database copy of the source device at the time of activation. For information on how PowerPath, ECA, and SRDF/A can be used for consistent split operations, refer to "TimeFinder consistent split" on page 167.

1. If the R2 is in a consistent state and the copy session was created with the `-precopy` option, specifying `-consistent` invokes SRDF/A to maintain consistency, instead of ECA.

Note: You can use the Enginuity Consistency Assist (ECA) feature to consistently activate copy sessions across multiple, heterogeneous hosts.

To consistently activate copy sessions using ECA, you must have either a control host with no database or a database host with a dedicated channel to the gatekeeper devices. In a SAN environment, gatekeepers and DATA devices may share the same FA port on the Symmetrix but the gatekeepers must be available on a separate host HBA than the DATA devices. This means that there must be a dedicated channel from a host HBA to the switch that can be used to access only gatekeepers and not devices that contain host data. This will ensure that in write intensive environments SYMAPI will be able to freeze and then thaw I/O to the devices in the device group within the ECA window, regardless of the number of outstanding I/Os held by the HBA.

Figure 2 illustrates how a control host can consistently activate a copy session involving three database hosts that access devices on a Symmetrix array.

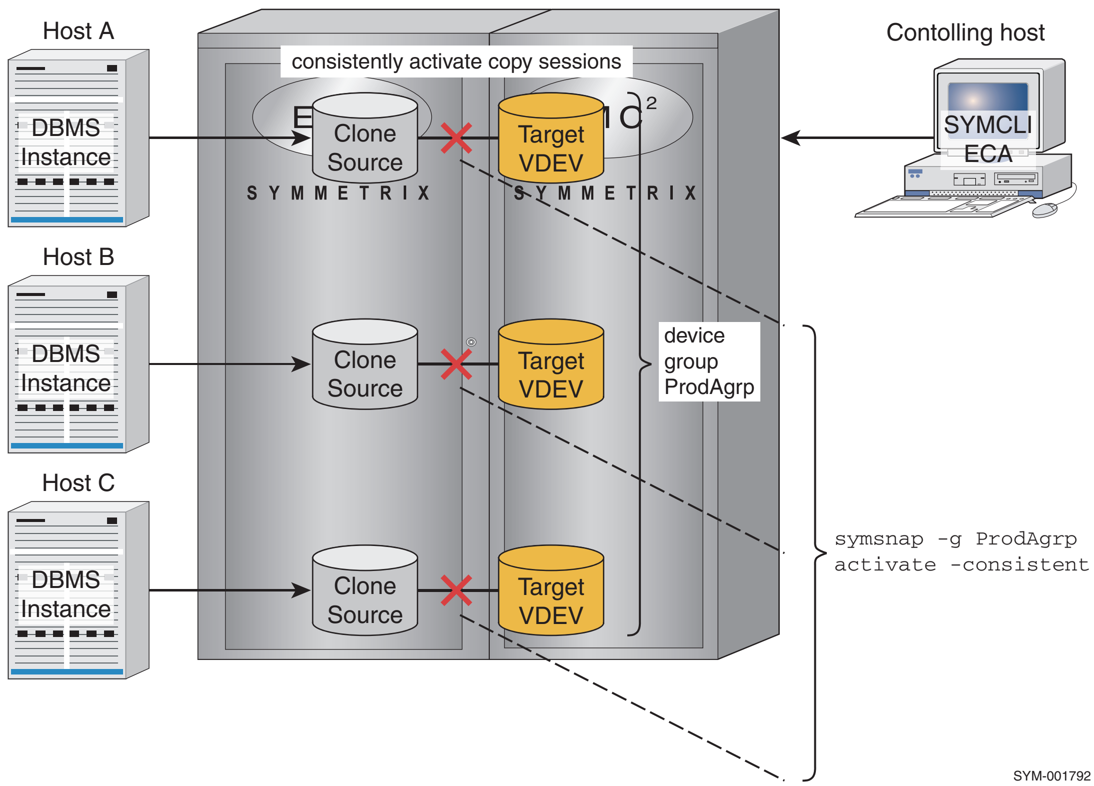

Figure 2 TimeFinder/Clone consistent activate using ECA

A Symmetrix device group, composite group, or a device file must be created on the controlling host for the target database to be consistently activated. Device groups can be created to include all of the devices being accessed or defined by database host access. For example, if you define a device group that includes all of the devices being accessed by Hosts A, B, and C (see Figure 2), then you can consistently activate all of the copy sessions related to those hosts with a single command. However, if you define a device group that includes only the devices accessed by Host A, then you can activate those copy sessions related to Host A without affecting the other hosts.

Activating a clone session for an additional pair in a group

When working with either a composite or device group, you can use the `-concurrent` option with the `activate` action to activate a copy session for an additional clone pair in a group.

Note: The copy session must exist prior to issuing the command.

For example, to activate a copy session for an additional clone pair in group `ProdDB`, enter:

```
symclone -g ProdDB activate -concurrent
```

Modifying a clone copy session

You can use the `set mode` argument to modify the mode in which a copy session is operating.

Note: With Enginuity Version 5x71 and later, you can use `set mode` to modify the mode between Copy, Nocopy, and Precopy on clone pairs that are in a Created, Recreated, or Activated state.

Possible values are:

- Copy — If a session was created without the `-copy` option, setting the mode to `copy` changes the session to Copy mode. That is, a copy will initiate once the session is activated, or the copy will start immediately if the session is already activated.
- Nocopy — If a session was created with the `-copy` option, setting the mode to `nocopy` changes the session to No-Copy mode. That is, the session will become CopyOnWrite/CopyOnAccess once the session is activated and no full device copy will initiate, or the copy will stop if the session is already activated.

  Note: Do not attempt to change a session created with the `-differential` option to the nocopy mode, as the session will fail.

- Precopy — If a session was created without the `-precopy` option, setting the mode to `precopy` will cause a Pre-Copy to take effect. Once you are in Pre-Copy mode, you can change to No-Copy mode. Once the session is activated, the session changes to Copy mode. For more information on Precopy, refer to "Starting to copy data before activating the session" on page 34.

For example, to modify the copy session created in the example on page 34 from Copy mode to Nocopy mode, enter:

```
symclone -g ProdDB set mode nocopy
```

Recreating a clone copy device

With Enginuity Version 5671 and later, once a clone device is fully copied, you can use the `symclone recreate` command to incrementally copy all subsequent changes made to the source device (made after the point-in-time copy initiated) to the target device.

To use this feature, the copy session must have been created with either the `-copy` or `-precopy` option, and the `-differential` option. In addition, the session must have been activated to establish the new point-in-time copy.

While in the Recreated state, the target device will remain Not Ready to the host.

For example, to recreate the copy session created in the example on page 34, enter:

```
symclone -g ProdDB recreate DEV001 sym ld DEV005
```

Starting to copy data before activating the session

The `-precopy` option can be used with the `recreate` argument to start copying tracks in the background, before the copy session is activated.

When using this option, a point-in-time copy will be established when the session is activated.

While in the Recreated state, the pre-copy process never actually completes. Instead, the process keeps checking for new writes to be pre-copied to the target device until the session is activated. Once activated, the normal background copy mechanism takes over and the pre-copy operation ceases.

You must use the `-precopy` option with the `recreate` argument if the session was initially created as a precopy session. You can also use the `-precopy` option with the `recreate` command on a session initially created with the `-copy` option.

Using the establish command

To recreate and then immediately activate a clone session with a single command, you can use the `symclone establish` command.

For example, to recreate and then activate the copy session shown in the example on page 34, enter:

```
symclone -g ProdDB establish DEV001 sym ld DEV005
```

This command implies the `-copy` and `-diff` options.

Note: The `symclone establish` command sets the target device to Not Ready for a short time. Therefore, you may want to unmount the target device before issuing the command.

In addition, you can use the `-concurrent` option with the `establish` command to activate an existing clone session for an additional clone pair in a group.

For example, to recreate and then activate a copy session for an additional clone pair in group `ProdDB`, enter:

```
symclone -g ProdDB establish -concurrent
```

Recreating a clone session for each pair in a group

When working with either a composite or device group, you can use the `-concurrent` option with the `recreate` action to recreate a clone session for each clone pair in a group.

For example, to recreate a copy session for each clone pair in group `ProdDB`, enter:

```
symclone -g ProdDB recreate -concurrent
```

Restoring data from a target device

You can use the `symclone restore` command to copy target data to another device (full restore), or back to the original source device (incremental restore).

In the case of a full restore (`-full`), the original session will terminate and a copy session to the target of the restore will start.

In the case of an incremental restore, the original session copy direction is reversed and changed data is copied from the target device to the source device. To support this operation, the session must have been created with the `-differential` option and the device must be in a fully copied state.

For example, to fully restore data from the original target device (`DEV005`) created in the example on page 34 to a device (`DEV006`) that was not involved in the original clone session, enter:

```
symclone -g ProdDB restore -full DEV006 sym ld DEV005
```

Note: When constructing a `symclone restore` command, the device receiving the data always appears first in the command, followed by the device from which the data is being copied. Therefore, in the above command, `DEV006` is actually the target of the data being copied from `DEV005`.

Splitting a clone device pair

With Enginuity Version 5671 and later, you can use the `symclone split` command to *split* a clone device pair that is in the Restored state. This command will change the direction of the clone relationship (that is, the original source device will become the source device for a future copy), which will enable you to use either the `establish` or `recreate` command.

For example, to split the pair created in the example on page 34, enter:

```
symclone -g ProdDB split DEV001 sym ld DEV005
```

Terminating a clone copy session

To terminate the copy session in the example in page 34 using the `symclone` command, enter:

```
symclone -g ProdDB terminate DEV001 sym ld DEV005
```

Terminating a copy session deletes the pairing information in the Symmetrix array and removes any hold on the target device.

Terminating a session while the device pairs are in the CopyOnAccess, CopyOnWrite, or CopyInProg state will cause the session to end. If the application has not finished accessing all of the data, the target copy is not a full copy.

The `symclone terminate` command is allowed for all TimeFinder/Clone pair states.

Note: A created and activated copy session may be terminated, but the data on the target device is not valid unless the state had previously been COPIED.

If the state is CopyInProg, then the `-symforce` option must be applied to terminate the session. But, this will also leave the target copy as an incomplete copy.

Using a clone from a clone target (both sessions are Clone)

In environments running Enginuity release level 5874 with Solutions Enabler V7.0, the target device of a clone session can be used as the source for one or more clone sessions. This allows a clone operation to take place with a device that is already involved in a clone operation without ending the first clone session. This applies to clone target devices of both TimeFinder/Clone sessions and TimeFinder/Clone Emulation sessions.

When clone sessions exist between multiple devices A, B, and C respectively, several configurations are possible:

- **Clone to clone target (A→B→C) session** — In this configuration, the source device A to target device B is a TimeFinder/Clone session, and source device B to target device C is a TimeFinder/Clone session (Figure 3),

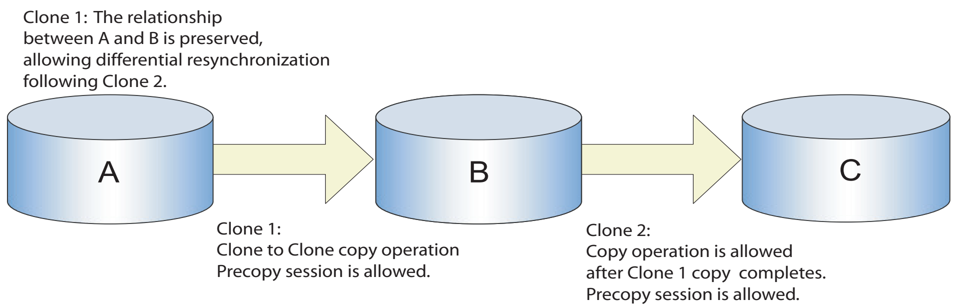

Figure 3 Clone from clone target (both sessions are cascaded clone)

As Figure 3 shows, cascaded sessions are accepted from left to right. This means you can use TimeFinder to clone device A to device B. Then, while the relationship between A and B is preserved, you can clone device B to device C.

Note: If you have session A→B→C, then session B→C can only be activated after session A→B has been activated. Precopy sessions are allowed.

- **Clone Emulation mode session to a clone session** — In this configuration, the source device A to target device B (BCV) is a TimeFinder/Clone Emulation session, and source device B to target device C is a TimeFinder/Clone session "Using cascaded Clone Emulation to cascaded clone" on page 140 describes this operation.
- **Clone session to a Clone Emulation mode session** — In this configuration, the source device A to target device B is a TimeFinder/Clone session, and source device B to target device C (BCV) is a TimeFinder/Clone Emulation session "Using cascaded Clone Emulation to cascaded clone" on page 140 describes this operation.

Note: Enginuity 5874 only allows a 2 hop (device A→ target device B→target device C) cascading relationship so long as the interactions rules listed on page 49 are followed. Any attempts to establish a 3 hop relationship (D→C when A→B→C or Z →A when A→B→C) will fail. Although circular cascading A→B→A is not allowed, devices A and B can have additional multiple targets. For example: A→B(1)→C(1) and A→B(2) and A→B(3)→C(2).

Interaction rules — When using a clone from a clone target, the following interaction rules apply:

- If the session state is `CopyInProg`, `SyncInProg` or `RestoreInProg`, the `-symforce` flag will be required.
- Recreate and incremental clone establish are allowed only on differential sessions.
- Set mode `nocopy` can only be done on nondifferential sessions.
- Set mode `precopy` can be done only in created and recreated states.

Note: The operations allowed for each configuration are not all the same and are dependent on the states of both relationships. The `Terminate` and `Cancel` operations are allowed for all session states.

Table 4 lists the configuration rules for using a clone from a clone target.

Table 4 **Clone from clone target session states**

B→C session state	Clone A → Clone B → Clone C session state					
	A→B No session	A→B Created Recreated	A→B Precopy	A→B CopyInProg CopyOnAccess CopyOnWrite	A→B Copied Split	A→B RestoreInProg	A→B Restored
B→C No session	Create A→B Full Establish A→B Full Restore A→B Create B→C Full Establish B→C Full Restore B→C	Activate A→B Set Mode A→B Create B→C	Activate A→B Set Mode A→B Create B→C (precopy)	Recreate A→B Establish A→B Set Mode A→B Create B→C	Recreate A→B Establish A→B Restore A→B Set Mode A→B Create B→C Full Establish B→C	Create B→C Full Establish B→C	Split A→B Create B→C Full Establish B→C Full Restore B→C
B→C Created Recreated	Create A→B (no precopy) Full Establish A→B Full Restore A→B Activate B→C Set Mode B→C	Activate A→B Set Mode A→B (no precopy) Set Mode B→C	Not proper state	Recreate A→B (no precopy) Establish A→B Set Mode A→B (no precopy) Set Mode B→C	Recreate A→B (no precopy) Establish A→B Restore A→B Set Mode A→B (no precopy) Activate B→C Set Mode B→C	Activate B→C Set Mode B→C	Split A→B Activate B→C Set Mode B→C
B→C Precopy	Create A→B Full Establish A→B Full Restore A→B Activate B→C Set Mode B→C	Activate A→B Set Mode A→B Set Mode B→C	Activate A→B Set Mode A→B	Recreate A→B Establish A→B Set Mode A→B Set Mode B→C	Recreate A→B Establish A→B Set Mode A→B Restore A→B Activate B→C Set Mode B→C	Activate B→C Set Mode B→C	Split A→B Activate B→C Set Mode B→C
B→C CopyInProg CopyOnAccess CopyOnWrite	Full Restore A→B Recreate B→C Establish B→C Set Mode B→C	Not proper state	Not proper state	Not proper state	Restore A→B Set Mode A→B (no precopy) Recreate B→C Establish B→C Set Mode B→C	Recreate B→C Establish B→C Set Mode B→C	Split A→B Recreate B→C Establish BC Set Mode B→C
B→C Copied Split	Create A→B Full Establish A→B Full Restore A→B Recreate B→C Establish B→C Restore B→C Set Mode B→C	Activate A→B Set Mode A→B Recreate B→C Set Mode B→C	Activate A→B Set Mode A→B Recreate B→C (precopy) Set Mode B→C	Recreate A→B Establish A→B Set Mode A→B Recreate B→C Set Mode B→C	Recreate A→B Establish AB Restore A→B Set Mode A→B Recreate B→C Establish B→C Set Mode B→C	Recreate B→C Establish BC	Split A→B Recreate B→C Establish BC Restore B→C
B→C RestoreInProg		Not proper state	Not proper state	Not proper state	Not proper state	Not proper state	
B→C Restored	Full Restore A→B Split B→C	Not proper state	Not proper state	Not proper state	Not proper state	Split B→C	Split B→C

Using a BCV as the clone source

As Figure 4 on page 51 shows, you can create a copy session between a BCV device and a target device. The controlling host performs I/O to the standard device that is established with a BCV as part of a BCV pair. At some point, when the BCV is synchronized with the standard device, you can split the BCV from the standard and create a copy session between the BCV and a target device that might be accessed by Host C. The split operation must be entirely complete, including the background phase, before you can create a copy session on it.

For additional information on using the `symmir` command and how to perform an instant split operation, refer to Chapter 4.

Pair states ruling clone operations

Because various other ongoing operations can conflict with your clone session, certain rules must be considered. The availability of some clone copy operations depends on the current state of SRDF and BCV pairs. The following rules apply to certain BCV pair states:

- If the source or target of a `symclone create` or `activate` operation is a BCV, the BCV pair state must be split. The split must be totally complete before the operation is allowed.
- Available with Enginuity 5874, a source device A to a target device B (BCV device) clone session is in Emulation Mode state and the source device B to target device C is a clone session.
- A TimeFinder standard device cannot be created or activated in a clone session if the BCV pair state is SplitBfrRest or RestInProg.
- The `symclone terminate` command is allowed for all TimeFinder BCV pair states.

Note: Appendix B explains the TimeFinder pair states that apply to TimeFinder/Clone copy sessions. Table 16 on page 211 lists For a description of each BCV pair state. "State rules for TimeFinder/Clone operations" on page 421 contains specific information regarding possible SRDF pair state conflicts.

Example: Creating a clone from a source device

The following example creates a *CopyOnAccess* copy session between source device `BCV001` in device group `ProdDB` and target device `DEV005` on the same Symmetrix array.

Once the copy session is activated, Host C can access target device tracks. If the accessed target tracks have not yet been copied, Symmetrix Enginuity software copies them for immediate access for Host C. If Host B writes to BCV device tracks that have not yet been copied, the Enginuity software immediately copies the tracks before allowing new data to overwrite those BCV tracks.

Note: In this example, where multiple hosts have access to the BCV source, you should consider using the `-not_ready` option with the `split` command to make the BCV not ready. This will enable you to keep the same data on the BCV and clone. If you decide to use this option, you may need to release any Not Ready state imposed on any devices once the session completes.

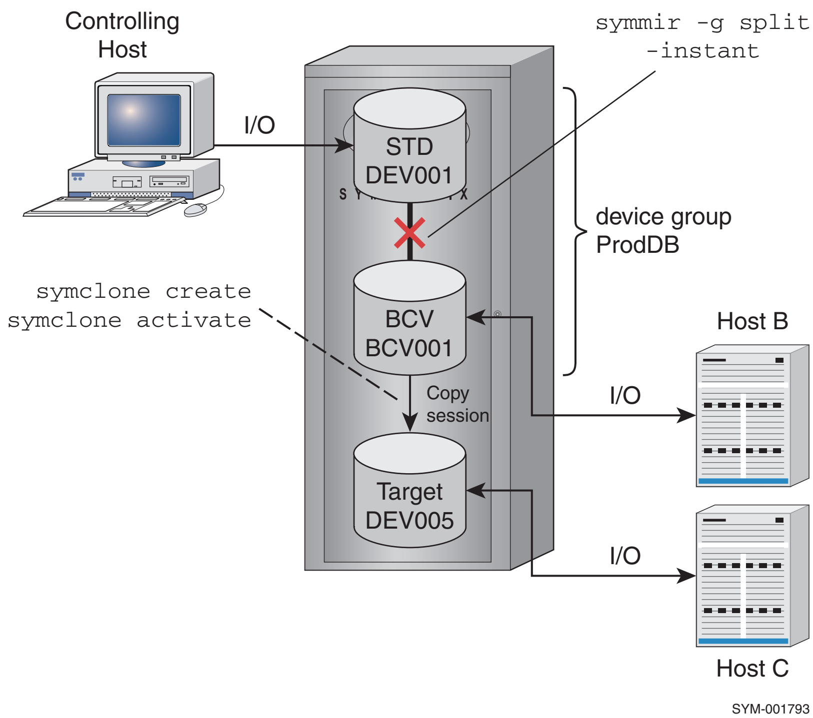

Figure 4 Creating a TimeFinder/Clone from a BCV source

The following steps outline the example shown in Figure 4:

1. Perform an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation:

   `symmir -g ProdDB split DEV001 -instant -not_ready -noprompt`

2. Verify that the background split is complete. The following command checks every 5 seconds:

   `symmir -g ProdDB verify DEV001 -split -bg -i 5`

3. Begin a copy session between the BCV source device `BCV001` and the standard target device `DEV005`:

   `symclone -g ProdDB create BCV001 sym ld DEV005`

4. Activate the copy session to Host C:

   `symclone -g ProdDB activate BCV001 sym ld DEV005`

5. Make the BCV device ready to its host:

   `symld -g ProdDB ready -bcv BCV001 -noprompt`

6. Query the state of the copy session and verify the CopyOnAccess state:

   `symclone -g ProdDB query`
   `symclone -g ProdDB verify BCV001 -copyonaccess`

7. When the application has finished accessing the data, the copy session and pair relationship can be terminated:

   `symclone -g ProdDB terminate BCV001 sym ld DEV005 -noprompt`

8. Incrementally reestablish the BCV pair:

   `symmir -g ProdDB establish DEV001 -noprompt`

Creating multiple clone copies

You can create up to 16 (non-differential) copies of a standard source device on various targets. A source device can be concurrently copying copies (`-copy` option) of the same data to up to eight target devices[1] at one time. For information on the number of clone copies that can be created, refer to "Understanding copy session limits" on page 32.

To create clone copies on the eight target devices, each target requires a separate copy session to become a clone. Target host devices cannot access the cloned data until the copy session has been activated. Up to 16 no-copy sessions can be activated at the same time on the same standard source device.

Creating multiple clone copies from a standard device

Figure 5 illustrates creating multiple clone copies from a standard source device `DEV001` on four standard target devices (`DEV005`, `DEV006`, `DEV007`, and `DEV008`) with various hosts accessing them.

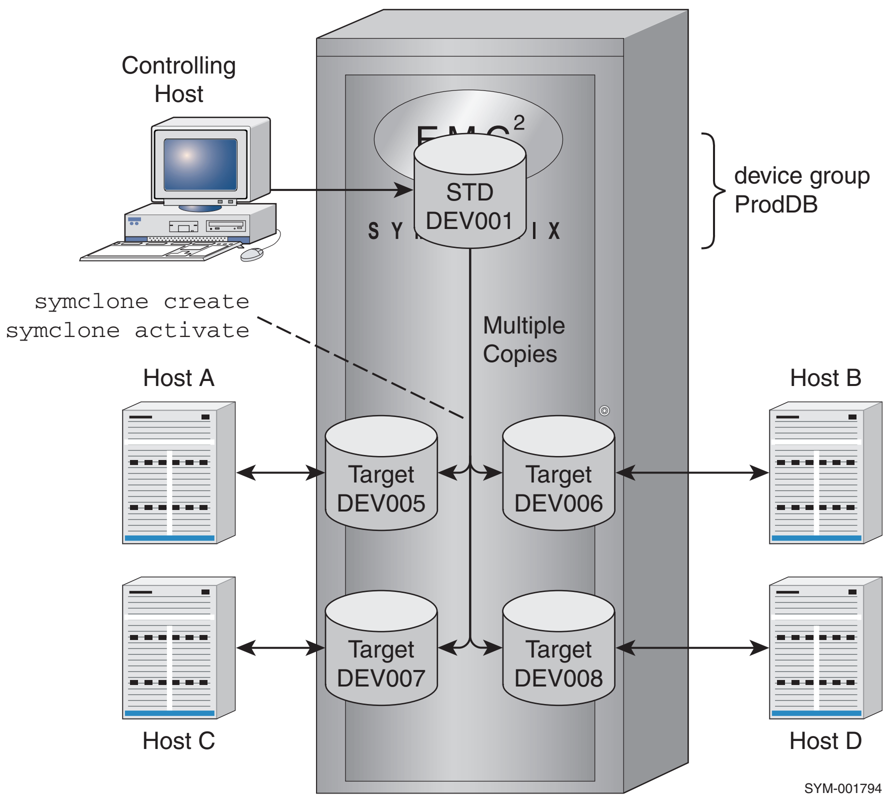

Figure 5 **Creating multiple clone copies from a standard device**

Note: A separate copy session must be created between the source device (`DEV001`) and each target device (`DEV005`, `DEV006`, `DEV007` and `DEV008`).

1. With Enginuity Version 5670, a source device can only concurrently copy to up to four target devices.

The following steps outline the example shown in Figure 5 on page 52:

1. Create a copy session between the standard source device `DEV001` and each of the four standard target devices `DEV005`, `DEV006`, `DEV007`, and `DEV008`. The `-copy` option is used to request background copying for a full device copy to the four target devices when the sessions are activated:

```
symclone -g ProdDB create DEV001 sym ld DEV005 -copy
symclone -g ProdDB create DEV001 sym ld DEV006 -copy
symclone -g ProdDB create DEV001 sym ld DEV007 -copy
symclone -g ProdDB create DEV001 sym ld DEV008 -copy
```

2. Activate the clone operation with one command to activate all copy sessions simultaneously:

```
symclone -g ProdDB activate DEV001 sym ld DEV005
                            DEV001 sym ld DEV006
                            DEV001 sym ld DEV007
                            BCV001 sym ld DEV008
```

3. Using the `query` argument you can display the state of all devices involved in the clone operation by including the `-multi` option or you can use the `verify` argument to verify that the data has been fully copied to the target:

```
symclone -g ProdDB query -multi
```

or:

```
symclone -g ProdDB verify DEV001 sym ld DEV005 -copied
symclone -g ProdDB verify DEV001 sym ld DEV006 -copied
symclone -g ProdDB verify DEV001 sym ld DEV007 -copied
symclone -g ProdDB verify DEV001 sym ld DEV008 -copied
```

4. When the full data copy has completed, the copy sessions and pair relationships can be terminated individually as needed:

```
symclone -g ProdDB terminate DEV001 sym ld DEV005 -noprompt
symclone -g ProdDB terminate DEV001 sym ld DEV006 -noprompt
symclone -g ProdDB terminate DEV001 sym ld DEV007 -noprompt
symclone -g ProdDB terminate DEV001 sym ld DEV008 -noprompt
```

Creating multiple clone copies from a BCV device

Figure 6 illustrates how to create multiple copies from a BCV source device BCV001 to four standard target devices (DEV005, DEV006, DEV007, and DEV008) with various hosts accessing them.

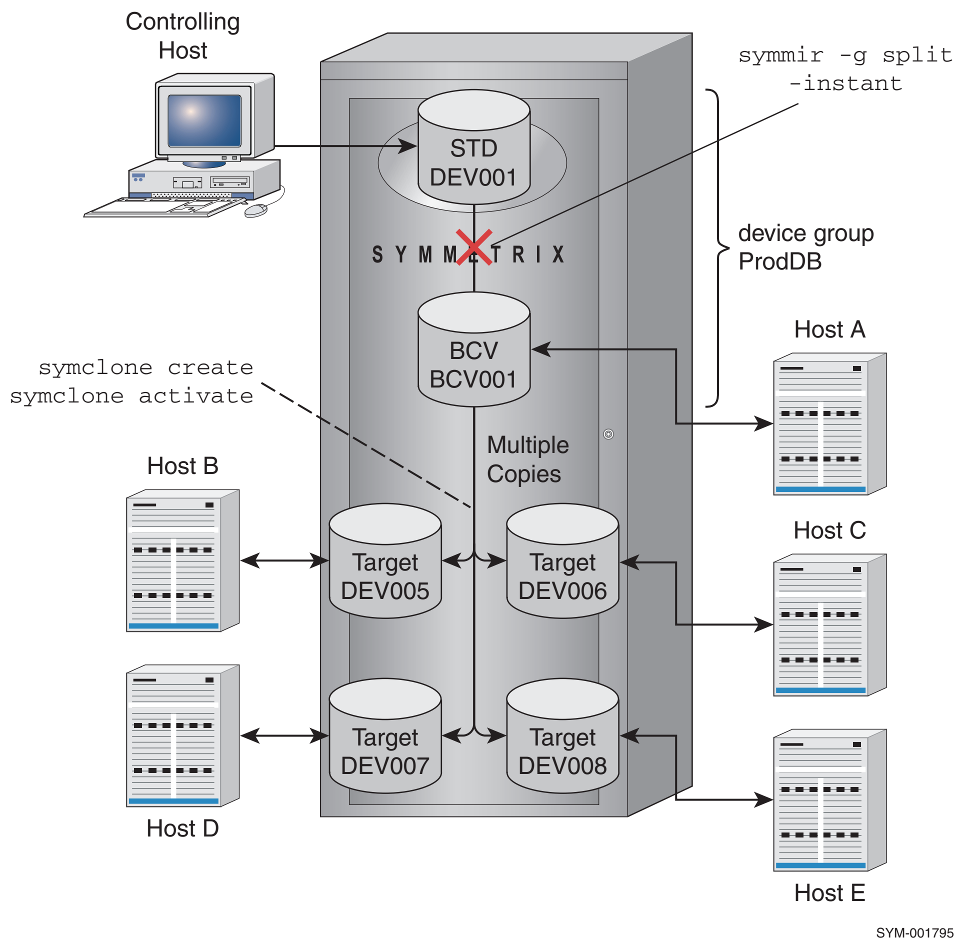

Figure 6 **Creating multiple clone copies from a BCV device**

The following steps outline the example shown in Figure 6:

1. Perform an instant split on the BCV pair. You can use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation:

```
symmir -g ProdDB split DEV001 -instant -not_ready -noprompt
```

Note: For additional information on using the `symmir` command and how to perform an instant split operation, refer to Chapter 4.

2. Verify that the background split is complete. The following command checks every 5 seconds:

```
symmir -g ProdDB verify DEV001 -split -bg -i 5
```

3. Create a copy session between the BCV source device `BCV001` and each of the four standard target devices `DEV005`, `DEV006`, `DEV007`, and `DEV008`. The `-copy` option is used to request background copying for a full device copy to the four target devices when the sessions are activated:

```
symclone -g ProdDB create BCV001 sym ld DEV005 -copy
symclone -g ProdDB create BCV001 sym ld DEV006 -copy
symclone -g ProdDB create BCV001 sym ld DEV007 -copy
symclone -g ProdDB create BCV001 sym ld DEV008 -copy
```

4. Activate the clone operation with one command to activate all copy sessions simultaneously:

```
symclone -g ProdDB activate BCV001 sym ld DEV005
                            BCV001 sym ld DEV006
                            BCV001 sym ld DEV007
                            BCV001 sym ld DEV008
```

Note: Because the copy sessions were created using the `-copy` option, the clone pair state will be CopyInProg when activated.

5. Make the BCV device ready to its host:

```
symld -g ProdDB ready -bcv BCV001 -noprompt
```

6. Using the `query` argument, you can display the state of all devices involved in the copy operation by including the `-multi` option or you can use the `verify` argument to verify that the data has been fully copied to the target:

```
symclone -g ProdDB query -multi
```

or:

```
symclone -g ProdDB verify BCV001 sym ld DEV005 -copied
symclone -g ProdDB verify BCV001 sym ld DEV006 -copied
symclone -g ProdDB verify BCV001 sym ld DEV007 -copied
symclone -g ProdDB verify BCV001 sym ld DEV008 -copied
```

7. When the full data copy has completed, the copy sessions and pair relationships can be terminated individually as needed:

```
symclone -g ProdDB terminate BCV001 sym ld DEV005 -noprompt
symclone -g ProdDB terminate BCV001 sym ld DEV006 -noprompt
symclone -g ProdDB terminate BCV001 sym ld DEV007 -noprompt
symclone -g ProdDB terminate BCV001 sym ld DEV008 -noprompt
```

8. Incrementally reestablish the BCV pair (`DEV001` and `BCV001`):

```
symmir -g ProdDB establish DEV001 -noprompt
```

Cloning a copy on a remote Symmetrix array

This section explains how to use SRDF to clone devices on a remote Symmetrix array.

Cloning a copy of a local R1 standard device on a remote Symmetrix array

Figure 7 illustrates how to clone a copy of a local R1 standard device on a remote Symmetrix array. Performing SYMCLI commands from the controlling host allows the remote target device to receive a copy of the data from the R2 device. The cloned data can be accessed by remote host A.

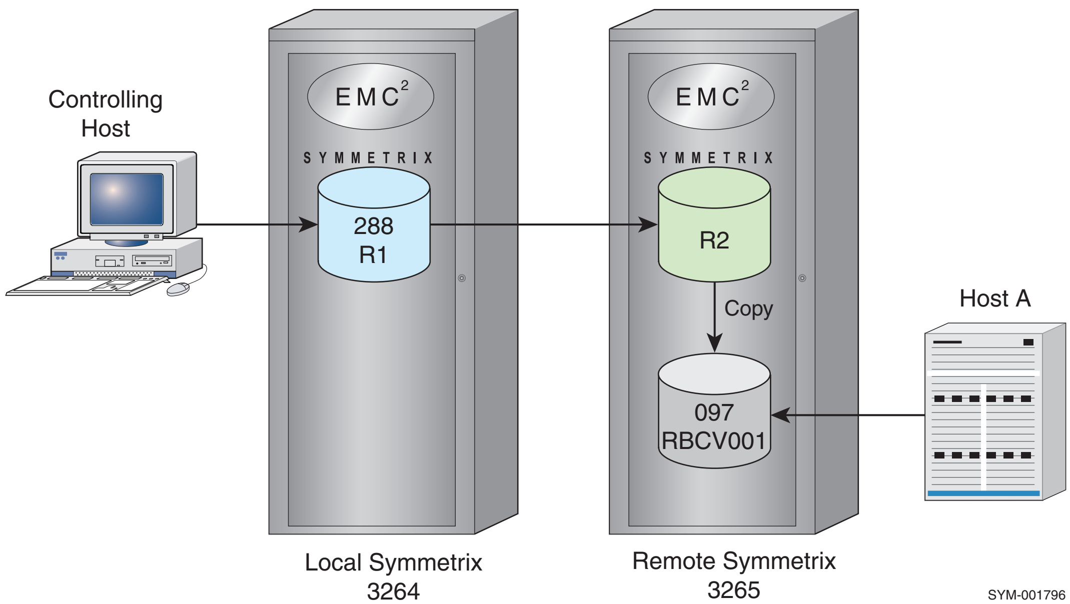

Figure 7 Cloning a copy of a local R1 standard device on a remote Symmetrix array

The following steps outline the example shown in Figure 7:

1. Create an RDF1 type device group:

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group an R1 standard device (288) on the local Symmetrix (sid 3264) to be the source device. Associate a target BCV device (097) on the remote Symmetrix to hold the clone copy. Note that to perform a remote operation you will need to use a remote target list (RTGT or RBCV list). This example uses an RBCV list.

```
symld -g Rdf1Grp -sid 3264 add dev 288
symbcv -g Rdf1Grp associate dev 097 -rdf
```

3. Clone an immediate full copy from the source device (DEV001) to the remote BCV target device (RBCV001). DEV001 is the logical device name for device 288, and RBCV001 is the logical device name for BCV 097:

```
symclone -g Rdf1Grp establish -full DEV001 bcv ld RBCV001 -rdf
```

4. To query the progress of the clone operation or verify when the copy is completed, issue the following commands that examine the clone pair (source and target):

```
symclone -g Rdf1Grp query -rdf
symclone -g Rdf1Grp verify -copied -rdf
```

Cloning a copy of a local BCV device on a remote Symmetrix array

Figure 8 illustrates how to clone a copy of a local BCV device on a remote Symmetrix array. Performing SYMCLI commands from the controlling host allows the remote target device to receive a copy of the data from the R2 device. The cloned data can be accessed by remote host A.

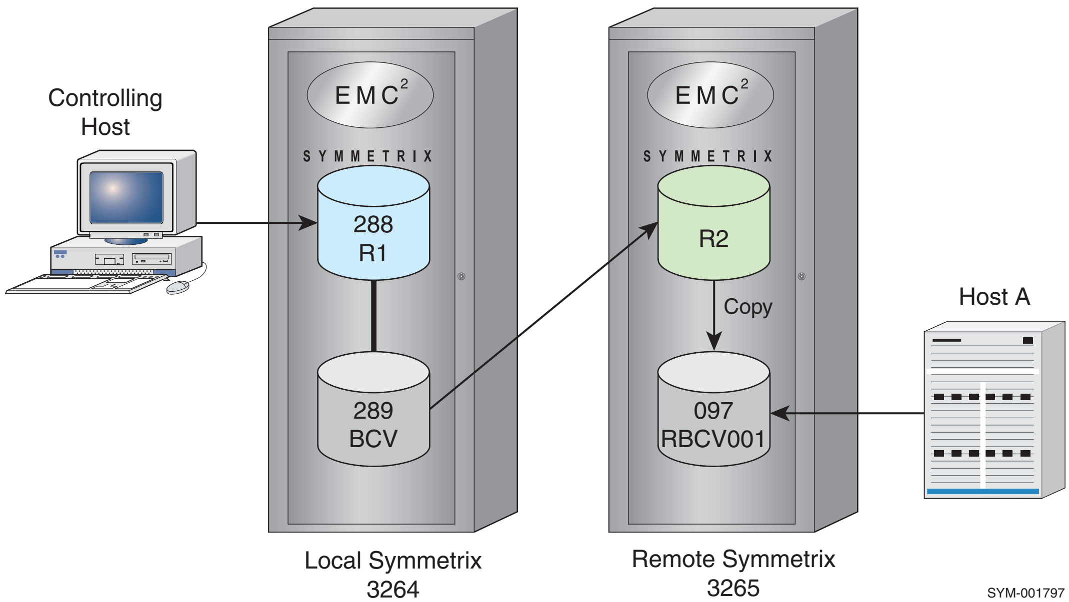

Figure 8 Cloning a copy of a local BCV device on a remote Symmetrix array

The following steps outline the example shown in Figure 8:

1. Create an RDF1 type device group:

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group a BCV device (289) on the local Symmetrix (sid 3264) to be the source device. Associate a target BCV device (289) on the local Symmetrix array. Associate a target BCV device (097) on the remote Symmetrix to hold the clone copy. In this case, the *remote* target device must be a BCV:

```
symld -g Rdf1Grp -sid 3264 add dev 288
symbcv -g Rdf1Grp -sid 3264 associate dev 289 -rdf
symbcv -g Rdf1Grp associate dev 097 -rdf
```

3. Clone an immediate full copy from the BCV device (BCV001) to the remote BCV target device (RBCV001). DEV001 is the logical device name for device 288, and RBCV001 is the logical device name for BCV 097:

```
symclone -g Rdf1Grp establish -full BCV001 bcv ld RBCV001 -rdf -bcv
```

4. To query the progress of the clone operation or verify when the copy is completed, issue the following commands that examine the clone pair (source and target):

```
symclone -g Rdf1Grp query -rdf -bcv
symclone -g Rdf1Grp verify -copied -rdf -bcv
```

Cloning a copy of a local R1 standard device on a hop-2 Symmetrix array

Figure 9 illustrates how to clone a copy of a local R1 standard device on a remote Symmetrix array in the second level of a multihop environment. Performing SYMCLI commands from the controlling host allows the remote target device to receive a copy of the data from the R2 device. The cloned data can be accessed by remote host A.

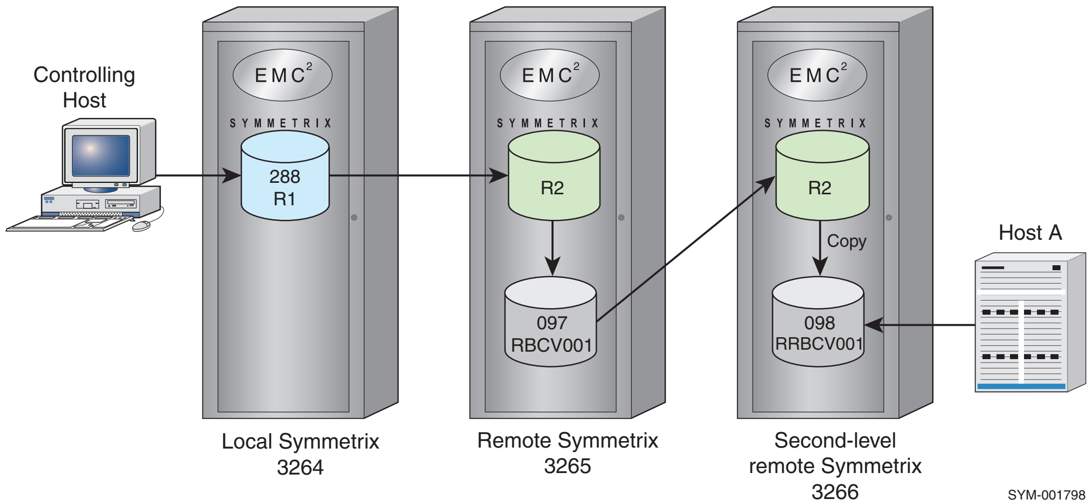

Figure 9 Cloning a copy of a local R1 standard device on a Hop-2 Symmetrix array

The following steps outline the example shown in Figure 9:

1. Create an RDF1 type device group:

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group an R1 standard device (288) on the local Symmetrix (sid 3264) to be the source device. Associate a BCV device (097) on the Hop-1 Symmetrix. Associate a BCV device (098) on the second-level (Hop-2) remote Symmetrix array to hold the clone copy. To perform a remote operation, you must use a remote target list (RTGT or RBCV list). Our example uses an RBCV list:

```
symld -g Rdf1Grp -sid 3264 add dev 288
symbcv -g Rdf1Grp associate dev 097 -rdf
symbcv -g RDF1GRP associate dev 098 -rrdf
```

3. Clone an immediate full copy from the source device (DEV001) to the remote BCV target device (RRBCV001). DEV001 is the logical device name for device 288, and RRBCV001 is the logical device name for BCV 098:

```
symclone -g Rdf1Grp establish -full RBCV001 bcv ld RRBCV001 -rrbcv
```

   Instead of specifying the device level, you can also specify the device group:

```
symclone -g Rdf1Grp establish -full -rrbcv
```

4. To query the progress of the clone operation or verify when the copy is completed, you can issue the following commands that examine the clone pair (source and target):

```
symclone -g Rdf1Grp query -rrbcv
symclone -g Rdf1Grp verify -copied -rrbcv
```

Cloning copies of the same data locally and remotely

Copies of the same data can be cloned to devices on a local and remote Symmetrix array at the same time so that their target devices have the same originator data.

Note that in the examples that follow, the procedures are very similar. The only differences are the device types used for the local copy (local target and local BCV target) and the command arguments used for the respective device type.

Example 1 Figure 10 illustrates how you can clone copies of a local R1 standard device to a local target device and to a remote BCV target device on a remote Symmetrix array.

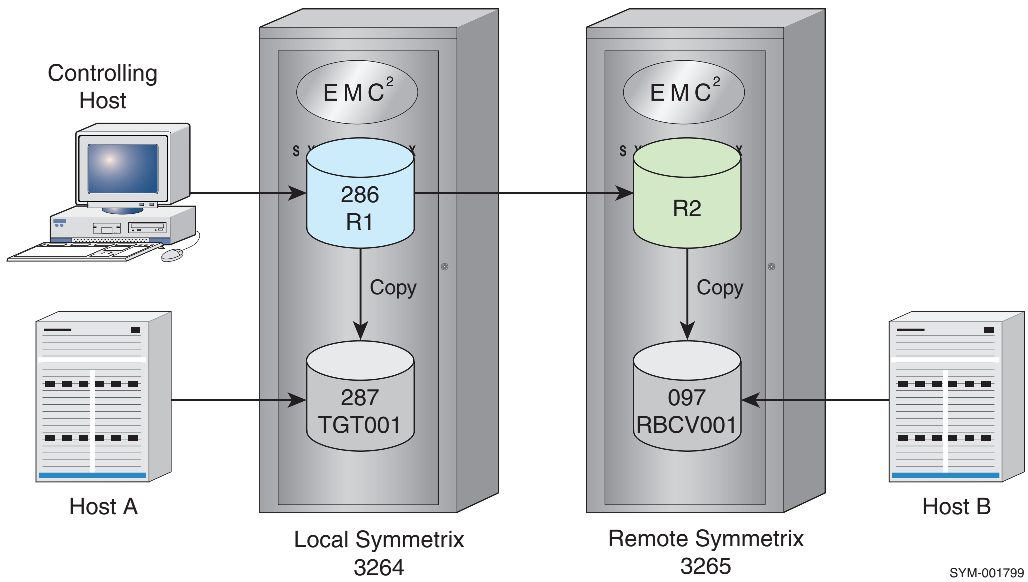

Figure 10 Cloning copies of a local R1 standard to a local target and to a remote BCV target

1. Create an RDF1 type device group:

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group an R1 standard device (286) on the local Symmetrix array (sid 3264) to be the source device. Add a standard device target (287) on the local array. Associate a remote BCV target (097) on the remote array. To perform a remote operation, you must use a remote target list (RTGT or RBCV list). This example uses an RBCV list:

```
symld -g Rdf1Grp -sid 3264 add dev 286
symld -g Rdf1Grp -sid 3264 add dev 287 -tgt
symbcv -g Rdf1Grp associate dev 097 -rdf
```

3. Clone an immediate full copy from the source device (DEV001) to the local and remote target devices (DEV002 and RBCV001, respectively). If there is no I/O to the source R1 device between these two commands, the same data will exist on both the local and remote target devices:

```
symclone -g Rdf1Grp establish -full -tgt
symclone -g Rdf1Grp establish -full -rdf
```

4. To query the progress of the local clone operation or verify when the local copy is completed, you can issue the following commands that examine the local cloned pair (source and target):

```
symclone -g Rdf1Grp query
symclone -g Rdf1Grp verify -copied
```

5. To query the progress of the remote clone operation or verify when the remote copy is completed, you can issue the following commands:

```
symclone -g Rdf1Grp query -rdf
symclone -g Rdf1Grp verify -copied -rdf
```

Example 2 Figure 11 illustrates how to clone copies of a local R1 standard device to a local BCV target device and to a BCV target device on a remote Symmetrix array.

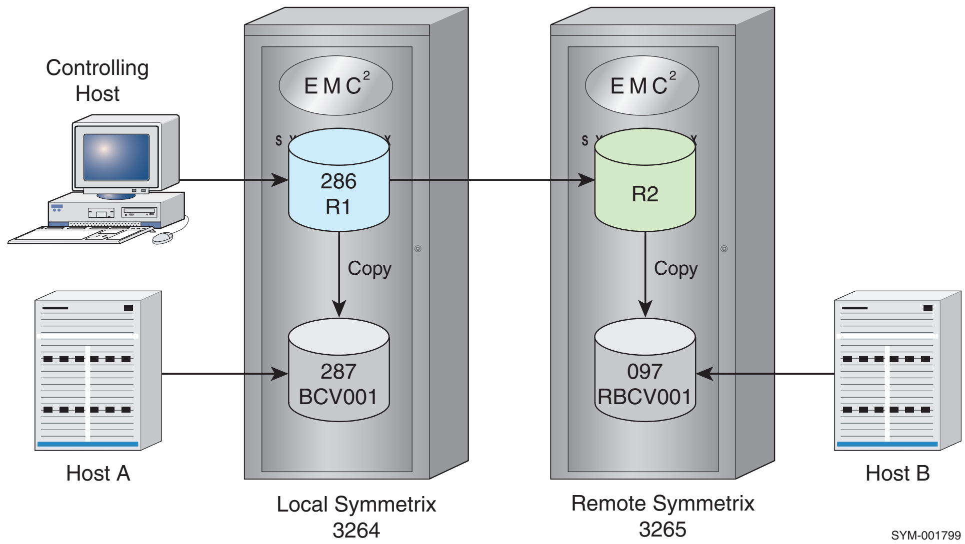

Figure 11 Cloning copies of a local device to a local BCV target and to a remote BCV target

1. Create an RDF1 type device group:

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group an R1 standard device (286) on the local Symmetrix array (sid 3264) to be the source device. Add a BCV target (287) on the local array. Associate a remote BCV target (097) on the remote array. To perform a remote operation, you must use a remote target list (RTGT or RBCV list). This example uses an RBCV list:

```
symld -g Rdf1Grp -sid 3264 add dev 286
symld -g Rdf1Grp -sid 3264 add dev 287 -tgt
symbcv -g Rdf1Grp associate dev 097 -rdf
```

3. Clone an immediate full copy from the source device (DEV001) to the local and remote target devices (BCV001 and RBCV001, respectively). If there is no I/O to the source R1 device between these two commands, the same data will exist on both the local and remote target devices:

```
symclone -g Rdf1Grp establish -full -tgt
symclone -g Rdf1Grp establish -full -rdf
```

4. To query the progress of the local clone operation or verify when the local copy is completed, you can issue the following commands that examine the local cloned pair (source and target):

```
symclone -g Rdf1Grp query
symclone -g Rdf1Grp verify -copied
```

5. To query the progress of the remote clone operation or verify when the remote copy is completed, you can issue the following commands:

```
symclone -g Rdf1Grp query -rdf
symclone -g Rdf1Grp verify -copied -rdf
```

Cloning multiple copies locally and remotely

Multiple copies of the same data can be cloned to devices on a local and remote Symmetrix array at the same time so that their target devices have the same originator data.

The configuration in Figure 12 is basically the same as the configuration in Figures 10 and 11 except that this configuration uses a single `symclone` command to clone copies from a source device to *four* target devices on each Symmetrix array instead of cloning copies to one target on each Symmetrix array. In this configuration, eight hosts have access to copies of the same target data.

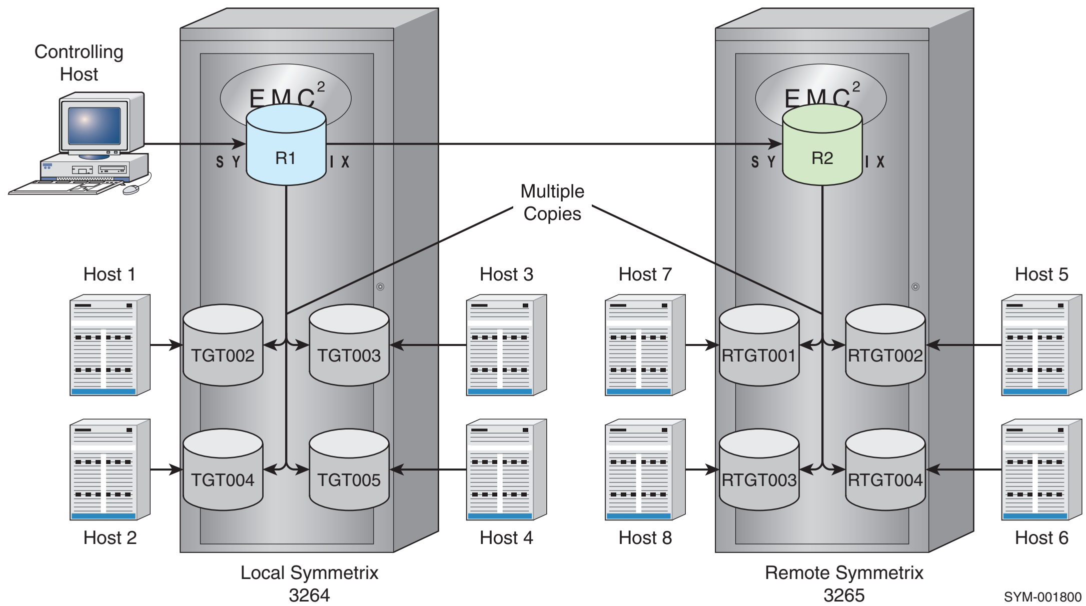

Figure 12 Cloning multiple copies on local and remote Symmetrix arrays

The process for building an RDF1 type device group is similar to the previous section. For example, to add a source standard R1 device, a range of four standard device targets from the local Symmetrix array, and a range of four targets from the remote Symmetrix array:

```
symld -g Rdf1Grp -sid 3264 add dev 100
symld -g Rdf1Grp -sid 3264 addall dev range 101:104 -tgt
symld -g Rdf1Grp addall dev range 214:218 -rdf -tgt
```

What is different about this configuration is the cloning of copies from a single source to multiple target devices. To clone the four *local* target devices (TGT002 – TGT005) from the local source device (DEV001), you need to issue four `symclone establish` commands, specifying the same source device with each of the four targets:

```
symclone -g Rdf1Grp establish -full DEV001 sym ld TGT002
symclone -g Rdf1Grp establish -full DEV001 sym ld TGT003
symclone -g Rdf1Grp establish -full DEV001 sym ld TGT004
symclone -g Rdf1Grp establish -full DEV001 sym ld TGT005
```

To display the progress of *all* devices involved in the *local* clone operation, perform a clone query with the `-multi` option:

```
symclone -g Rdf1Grp query -multi
```

You can verify the clone completion of one or more clone pairs in the device group specifically, or all clone devices. For example:

```
symclone -g Rdf1Grp verify -copied DEV001 sym ld DEV002
symclone -g Rdf1Grp verify -copied
```

To clone the four remote target devices (RTGT001 – RTGT004) from the source device (DEV001), you need to issue four `symclone establish` commands with the `-rdf` option:

```
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RTGT001
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RTGT002
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RTGT003
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RTGT004
```

To display the progress of all devices involved in the *remote* clone operation, you need to perform a clone query with the `-multi` option and the `-rdf` option:

```
symclone -g Rdf1Grp query -multi -rdf
```

You can verify the clone completion of one or more remote clone pairs in the device group specifically, or all remote clone devices, by adding the `-rdf` option. For example:

```
symclone -g Rdf1Grp verify -copied -rdf DEV001 sym ld RTGT004
symclone -g Rdf1Grp verify -copied -rdf
```

Cloning a copy at the tertiary site of a Cascaded SRDF configuration

Using SRDF technology and the TimeFinder hop2 flag (`-hop2` option), you can clone devices on a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration (Figure 13). Performing SYMCLI commands from the controlling host allows the tertiary target device to receive a copy of the data from the R2 device. The cloned data can be accessed by remote host A.

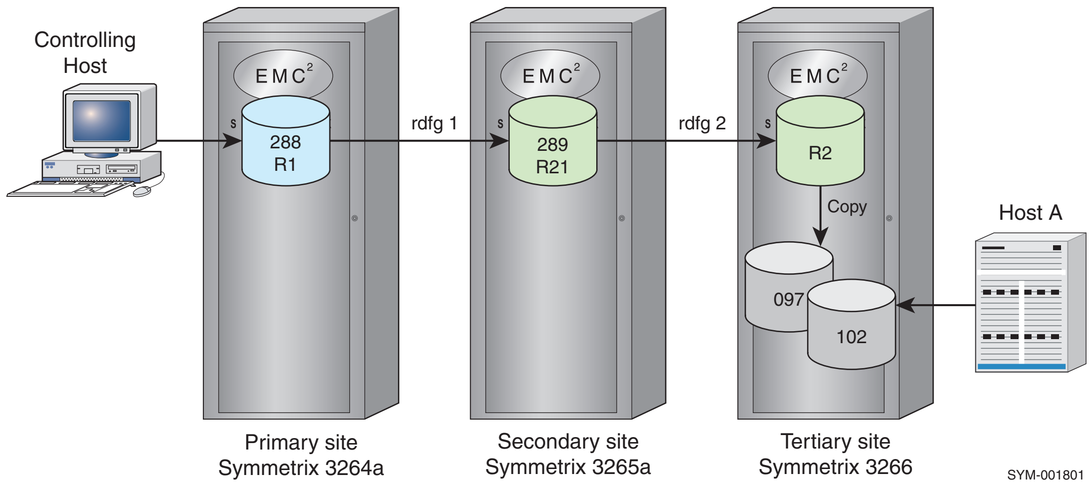

Figure 13 Cloning a copy at the tertiary site of a Cascaded SRDF configuration

The following steps outline an example of cloning devices on a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration:

Note: The following procedure provides examples of device group and composite group commands.

---

1. Create an RDF1-type device group or composite group (for example, a group named `RDF1Grp`):

   To create an RDF1-type device group:

   ```
   symdg create Rdf1Grp -type rdf1
   ```

   To create an RDF1-type composite group:

   ```
   symcg create Rdf1Grp -type rdf1
   ```

2. Add devices to the group. From the primary site Symmetrix, add an R1 standard device to be the source device. From the tertiary site Symmetrix, add a target BCV device (097) to hold the clone copy:

   To add devices to a device group:

   ```
   symld -g Rdf1Grp -sid 3264 add dev 288
   symbcv -g Rdf1Grp -hop2 -rdfg 1 -remote_rdfg 2 add dev 097
   ```

   To add devices to a composite group:

   ```
   symcg -cg Rdf1Grp -sid 3264 add dev 288
   symbcv -cg Rdf1Grp -sid 3264 -hop2 -rdfg 1 -remote_rdfg 2 add dev 097
   ```

To add target devices to a device group using the `-tgt` option:

```
symld -g Rdf1Grp -hop2 -rdfg 1 -remote_rdfg 2 -tgt add dev 102
```

To add devices to a composite group using the `-tgt` option:

```
symld -cg Rdf1Grp -sid 3264 -hop2 -rdfg 1 -remote_rdfg 2 -tgt add dev 102
```

3. Clone an immediate full copy from the source device to the remote BCV target device:

```
symclone -g Rdf1Grp -hop2 create -precopy -diff
```

Or, you can use the `-tgt` option:

```
symclone -g RdfGrp1 -hop2 create -precopy -diff -tgt
```

4. To query the progress of the clone operation or verify when the copy is completed, you can issue the following commands that examine the clone pair (source and target):

```
symclone -g Rdf1Grp query -hop2
symclone -g Rdf1Grp verify -copied -hop2
```

Using composite groups to manage clone pairs across Symmetrix arrays

Figure 14 shows a production host locally connected to two Symmetrix arrays (A and B). A composite group is defined on the production host and includes source devices and target devices from each Symmetrix array. The target devices can be standard devices or BCV devices. Another locally connected host allows access to the clone targets.

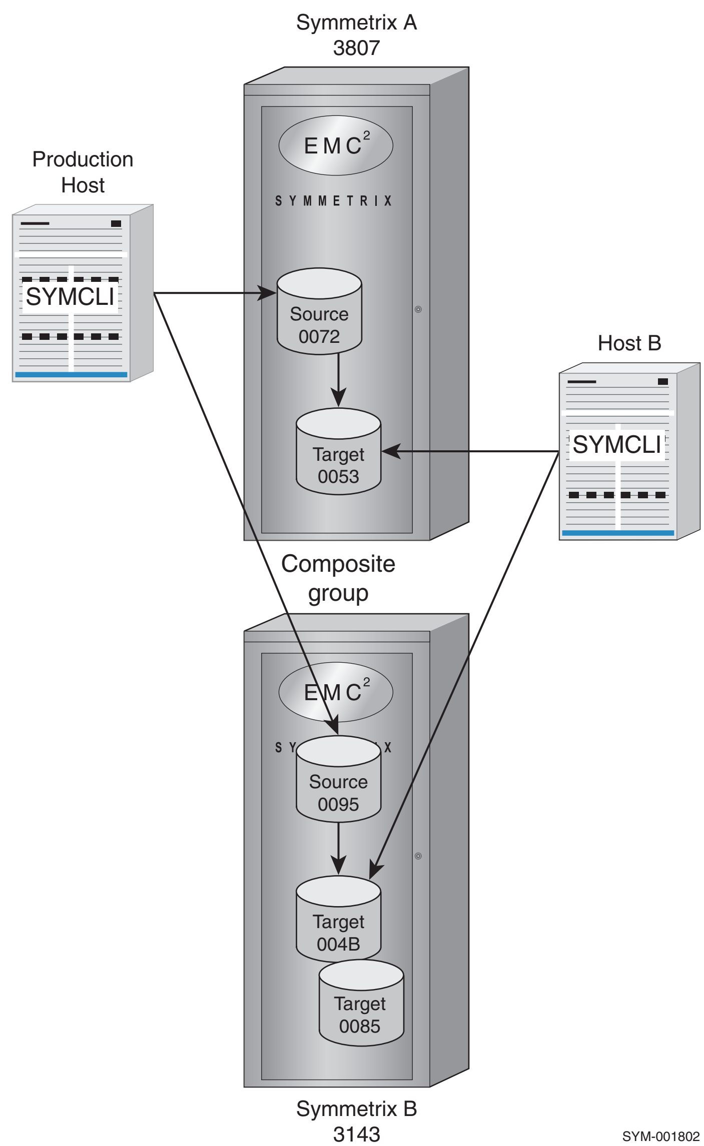

Figure 14 Using a composite group when a set of devices spans two Symmetrix arrays

Although clone copy operations might normally be performed from the production host (Figure 14) because the composite group is defined there in its SYMAPI database, there are methods that would allow you to control clone operations from another locally connected host like the target host. One way is to copy the composite group definition to another host. A more efficient method is to enable Group Naming Services (GNS), which automatically propagates the composite group definition to the Symmetrix arrays and other locally attached hosts that are running the GNS daemon. For more information, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

The following steps explain how to setup a composite group that spans two Symmetrix arrays as shown in Figure 14 on page 65:

1. From the production host, create a Regular type composite group (for example, `MyGrp`):

   ```
   symcg create MyGrp -type regular
   ```

2. Add to the composite group those standard devices on Symmetrix A (`3087`) and Symmetrix B (`3143`) that are the source devices:

   ```
   symcg -cg MyGrp -sid 3087 add dev 0072
   symcg -cg MyGrp -sid 3143 add dev 0095
   ```

3. Associate a BCV target device from each Symmetrix array with the composite group:

   ```
   symbcv -cg MyGrp -sid 3087 associate dev 0053
   ```

4. Create clone pair sessions from those devices in the composite group:

   ```
   symclone -cg MyGrp create
   ```

5. Activate these clone pair sessions:

   ```
   symclone -cg MyGrp activate
   ```

Once you have setup the composite group, you can control specific clone pairs within it, as long as the devices reside in the same Symmetrix array. For example, to create and activate only the DEV001/DEV002 clone pair from all devices in the group:

```
symclone -cg MyGrp create DEV001 sym ld DEV002
symclone -cg MyGrp activate DEV001 sym ld DEV002
```

Or using the one-step method:

```
symclone -cg MyGrp establish -full DEV001 sym ld DEV002
```

Command options with device groups

Table 5 lists the `symclone` control operations and the possible options to use when targeting a specified device group.

Table 5 symclone -g control arguments and possible options (page 1 of 2)

	Argument action										
Options	create	activate	establish -full	establish	recreate	split	restore -full	restore	terminate	query	verify
`-bcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-both_sides`		✔	✔	✔							
`-c`, `-i`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-concurrent`	✔	✔	✔	✔	✔						✔
`-consistent`		✔	✔	✔							
`-copied`											✔
`-copy`	✔										
`-copyinprog`											✔
`-copyonaccess`											✔
`-copyonwrite`											✔
`-created`											✔
`-diff`	✔										
`-exact`	✔		✔				✔	✔			
`-force`	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔
`-hop2`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-multi`										✔	
`-noprompt`	✔	✔	✔	✔	✔	✔	✔	✔	✔		
`-not_ready`		✔	✔	✔			✔	✔			
`-offline`										✔	✔
`-opt`	✔		✔								
`-ppath`		✔	✔	✔							
`-preaction`, `-postaction`		✔	✔	✔							
`-precopy`	✔				✔						✔
`-preservetgtlocks`, `-lockid`	✔	✔	✔	✔	✔	✔	✔	✔	✔		
`-rbcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-rrbcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Table 5 symclone -g control arguments and possible options (page 2 of 2)

	Argument action										
Options	create	activate	establish -full	establish	recreate	split	restore -full	restore	terminate	query	verify
`-rdb`, `-db`, `-dbtype`		✔									
`-rdf`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-skip`	✔	✔			✔	✔			✔		
`-star`	✔	✔	✔	✔	✔	✔	✔	✔	✔		
`-symforce`									✔		
`-tgt`	✔	✔	✔	✔	✔	✔	✔	✔	✔		
`-v`	✔	✔	✔	✔	✔	✔	✔	✔	✔		
`-vxfs`		✔									

a. Set by default.

Command options with composite groups

Table 6 lists the `symclone` control operations and the possible options to use when targeting a specified composite group.

Table 6 symclone -cg control arguments and possible options (page 1 of 2)

	Argument action									
Options	**create**	**activate**	**establish -full**	**establish**	**recreate**	**restore -full**	**restore**	**terminate**	**query**	**verify**
`-bcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-c`, `-i`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-concurrent`	✔	✔	✔	✔	✔					✔
`-consistent`		✔	✔	✔						
`-copied`										✔
`-copy`	✔									
`-copyinprog`										✔
`-copyonaccess`										✔
`-copyonwrite`										✔
`-created`										✔
`-diff`	✔									
`-exact`	✔		✔			✔	✔			
`-force`	✔	✔	✔	✔	✔	✔	✔	✔		
`-hop2`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-multi`									✔	
`-noprompt`	✔	✔	✔	✔	✔	✔	✔	✔		
`-not_ready`		✔	✔	✔		✔	✔			
`-offline`									✔	✔
`-opt`	✔		✔							
`-opt_rag`	✔		✔							
`-ppath`		✔	✔	✔						
`-preaction`, `-postaction`		✔	✔	✔						
`-precopy`	✔				✔					✔
`-preservetgtlocks`, `-lockid`	✔	✔	✔	✔	✔	✔	✔	✔		
`-rbcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-rrbcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-rdb`, `-db`, `-dbtype`		✔								
`-rdf`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Table 6 symclone -cg control arguments and possible options (page 2 of 2)

	Argument action									
Options	create	activate	establish -full	establish	recreate	restore -full	restore	terminate	query	verify
-skip	✔	✔						✔		
-star	✔	✔	✔	✔	✔	✔	✔	✔		
-symforce								✔		
-tgt	✔	✔	✔	✔	✔	✔	✔	✔		
-v	✔	✔	✔	✔	✔	✔	✔	✔		
-vxfs		✔								

a. Set by default.

Command options with device files

Table 7 lists the `symclone` control operations and the possible options to use when targeting device pairs specified in a device file of a given Symmetrix array.

Table 7 symclone -file control arguments and possible options

	Argument Action							
Options	**create**	**activate**	**establish**	**recreate**	**restore**	**terminate**	**query**	**verify**
`-both_sides`		✔	✔					
`-c`, `-i`	✔	✔	✔		✔	✔	✔	✔
`-consistent`		✔	✔					
`-copied`								✔
`-copy`	✔							
`-copyinprog`								✔
`-copyonaccess`								✔
`-copyonwrite`								✔
`-created`								✔
`-diff`	✔							
`-force`	✔	✔	✔	✔	✔	✔		
`-multi`							✔	
`-noprompt`	✔	✔	✔	✔	✔	✔		
`-not_ready`		✔	✔		✔			
`-ppath`		✔	✔					
`-preaction, -postaction`		✔	✔					
`-precopy`	✔			✔				✔
`-preservetgtlocks, -lockid`	✔	✔	✔	✔	✔	✔		
`-rdb`, `-db`, `-dbtype`		✔						
`-sid`	✔	✔	✔	✔	✔	✔	✔	✔
`-skip`	✔	✔				✔		
`-star`	✔	✔	✔	✔	✔	✔		
`-symforce`						✔		
`-v`	✔	✔	✔	✔	✔	✔		
`-vxfs`		✔						

a. Set by default.

3

Performing TimeFinder/Snap Operations

This chapter describes how to control copy sessions for virtual devices using the SYMCLI `symsnap` command.

TimeFinder/Snap environment

The SYMCLI TimeFinder/Snap operations enable you to create and manage virtual copy sessions between a source device and multiple *virtual* (VDEV) target devices. This functionality requires a TimeFinder/Snap or EMC Snap license. The source device can be either a standard device or a BCV device. The target device(s) must be virtual device(s). For a description of *virtual* devices and their associated *SAVE* devices, refer to Chapter 1.

Copies of striped or concatenated metadevices can also be created as long as the source and target virtual metadevices are completely identical in stripe count, stripe size, and capacity. Once the copy session has been activated, the device copy can be accessed via the virtual target device instantly.

TimeFinder/Snap for virtual device copy operations provides multiple copies of production data for testing, backups or report generation. It can also be used to reduce disk contention and improve data access speed, by assigning users to copies of data rather than accessing the one production copy. Depending on whether or not a source device has other sessions, a single source device may have as many as fifteen copies. "Understanding copy session limits" on page 76 contains more information on the number of available device copies

Virtual snap copies can prove useful in situations where multiple backups or recovery copies of a source device are taken throughout the day, with only a small percentage of data actually changing on the device. As an alternative to using multiple BCVs to backup these devices, you can use virtual devices to reduce costs and save on storage space. Through the use of device pointers to the original data, virtual devices allow you to allocate space based on the expected amount of the changes to a device, rather than the size of the volume being backed up.

Snap operations overview

TimeFinder/Snap functionality is managed through copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix array and can be queried to verify the current state of the devices. A copy session must first be created that defines the snap devices in the copy operation. Once the session is subsequently activated, the target virtual devices then become accessible to its host. When the information is no longer needed, the session can be terminated. In addition, data can also be restored from the virtual devices back to the source devices.

Snap operations are controlled from the host by using the `symsnap` command to `create`, `activate`, `terminate`, and `restore` the snap copy sessions. The snap operations described in this chapter explain how to manage the devices participating in a copy session using the SYMCLI.

Figure 15 illustrates a virtual copy session where the controlling host creates a copy of standard device `DEV001` on target device `VDEV005`.

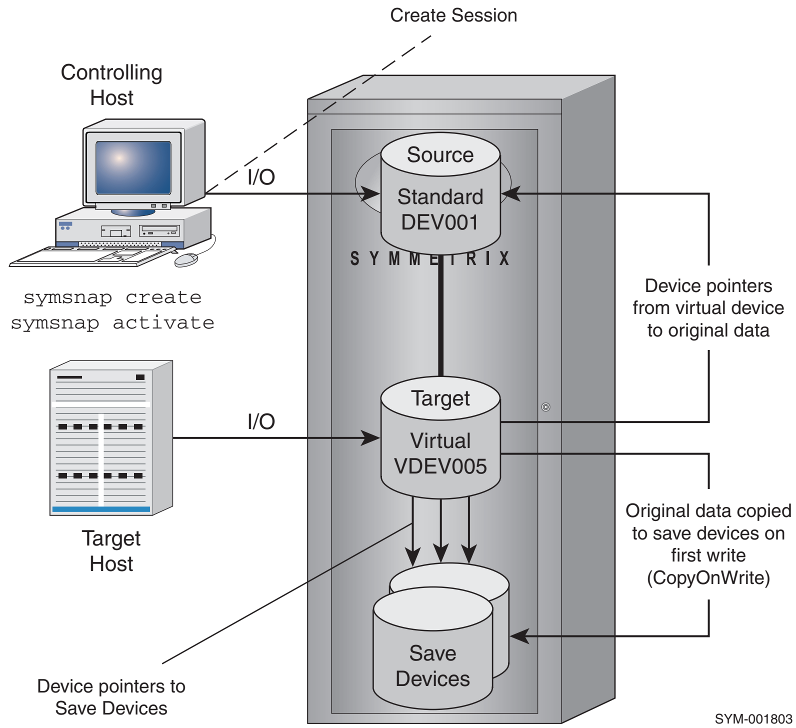

Figure 15 Copy of a standard device to a virtual device (VDEV)

As illustrated in Figure 15, a virtual device is a host-accessible device that contains a collection of pointers to unchanged data and to save data. When you activate a snap pair, disk space is consumed (on SAVE devices) only when data is written to the source device or VDEV, and then only the space required to accommodate pre-update data from the source tracks that changed.

Pointers on the virtual device are initialized to point to tracks on the source device. The first time new data is written to a track on the source device, the original track data is copied to a SAVE device, and the pointer on the virtual device is changed to point to that original data. When a track is written to a VDEV, it also gets copied to the SAVE device and the pointer on the VDEV is changed to point to the SAVE device.

Creating a virtual copy session

Initially, you must create a virtual copy session that defines and sets up the snap devices you have selected for the snap operation. For example, to begin a copy session and define a specified target device `VDEV005` to be the copy of source device `DEV001` in group `ProdDB`, enter:

```
symsnap -g ProdDB create DEV001 vdev ld VDEV005
```

The `symsnap create` action defines the copy session requirements and sets the track protection bitmap on the source device to protect all tracks and detect which tracks are being accessed by the target host or written to by the source host. The target virtual device remains Not Ready to its host and placed on hold status for copy session usage. This prevents other control operations from using the device. The device pair state will transition from CreateInProg to Created when complete. The virtual data becomes accessible to its host when the copy session is activated. Refer to "Activating a virtual copy session" on page 80 for more information.

Understanding copy session limits

The Symmetrix array is currently limited to 16 sessions per source, STAR, or ORS device, which can be used for TimeFinder or SDDF operations. This limits the number of available clone copies that can be created. Although a total of 16 concurrent CopyOnAccess copy sessions can be created from a standard device, one copy session is reserved for Restore operations. This number is decremented for each current TimeFinder/Clone, TimeFinder/Snap, TimeFinder/Mirror, and SDDF session. For example:

- When a standard device is two paired BCV devices, only 14 copy sessions can be created for that standard device.
- When running in TimeFinder/Clone Emulation mode, this number is decremented an additional two sessions per single BCV relationship.

Copy sessions created using the `-copy` option to create full data copies are limited to eight concurrent sessions, with up to an additional eight CopyOnAccess sessions, for a total of 16 sessions.

Note: Depending on the specific level of Enginuity that you are running, snap restore operations using the `symsnap restore` command can create an additional copy session per the source device. For information on restoring virtual devices, refer to "Restoring data from virtual devices" on page 82.

Multivirtual snaps

Solutions Enabler supports up to 128 snaps from a source device. This support requires that you enable the following SYMCLI environment variable:

```
SYMCLI_MULTI_VIRTUAL_SNAP = ENABLED
```

This setting appears in the snap device output from `symdev show`. The **Snap State Flags** value will be `MultiVirtual`.

A single source device can only have snaps of one type.

Note: Multivirtual snaps are not supported with the Auto Cancel policy, CKD devices, and thin standard devices. For environments running Enginuity release level 5874 and later, Snap Recreate is not supported with multivirtual snaps.

Copying data on first write

When the copy session is activated, a point-in-time copy of the data becomes available to a target connected host. The state of the device pair is CopyOnWrite. This means that any first write to the source device or target device during the active copy session will result in the data being copied to an additional SAVE device for storage. The target host accesses the copy through the use of device pointers stored on the virtual device. The pointers keep track of where the data is located in physical storage for accessing the point-in-time copy from the time of activation.

Figure 15 on page 75 illustrates a virtual copy session where the controlling host creates a copy of `DEV001` on a virtual target device `VDEV005`.

Note: For a detailed description of virtual devices and their associated SAVE devices, refer to Chapter 1.

If data is written to a track on the source device for the first time after a copy session is activated, the original data is copied to the SAVE device before overwriting the source and updating the pointers to the data on the virtual device. If data is written to the target device after the copy session is activated, that data is immediately stored on the SAVE devices. Once the original track data has been changed, the track will remain unprotected on the source.

Optionally, data can be restored to another target device before terminating the copy session. For information on restoring a virtual device to another device, refer to "Restoring data from virtual devices" on page 82.

Note: For a comparison of copy functionality between TimeFinder/Clone operations (`symclone`) for standard and BCV devices and TimeFinder/Snap operations (`symsnap`) for virtual devices, refer to Table 9 on page 110.

Specifying a SAVE device pool

Symmetrix supports the creation of multiple named SAVE device pools, allowing `symsnap` commands to use a particular pool. For more information on SAVE device pools, refer to "SAVE device" on page 21.

The `-svp` option can be used with the `create` action to specify which SAVE device pool to use for an operation. For example, to instruct the copy session created in the example on page 76 to use the SAVE device pool `Accounting`, use the following `symsnap` command:

```
symsnap -g ProdDB create DEV001 vdev ld VDEV005 -svp Accounting
```

Note: The specified pool must exist and contain at least one enabled device before creating the copy session.

In addition to the `-svp` option, you can also set the environment variable SYMCLI_SVP to a poolname to be used when `-svp` is not present in the command line. If `-svp` and SYMCLI_SVP are not used, the operation will use the default pool, DEFAULT_POOL.

Note: SAVE devices can also be organized into *Delta Set Extension* pools for use with SRDF/A. For information on using SAVE devices in this manner, refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide*.

Monitoring SAVE device usage

Using virtual copies requires proper planning to prevent the SAVE devices from filling up with pre-updated data. If the SAVE devices fill up, you will begin to lose the pre-update images of the newly changed tracks in the virtual copy session, the virtual device will be set to Not Ready, and the session will fail (only sessions with I/O activity will be in a failed state, sessions without I/O activity will continue to operate normally). Should this happen, you must terminate the failed sessions to clear the tracks on the SAVE devices. Once a session is terminated, the virtual data is lost and the SAVE device space associated with the session is freed and returned to the SAVE device pool for TimeFinder/Snap use. In addition, you should also examine how you are using the SAVE device pools and consider adding more SAVE devices.

You can monitor SAVE devices by using the `symsnap monitor` command to check the percentage full. When devices reach the specified percentage, an optional action script can be executed by the application to preserve the data or terminate sessions. The following is an example of the monitor command:

```
symsnap monitor -percent 80 -action SaveScript -norepeat -i 60 -svp
Accounting
```

Note: You can also use the `symcfg show` command to display SAVE device pool details. For more information, refer to the *Solutions Enabler Symmetrix Array Management CLI Product Guide*.

In this example, the SAVE device pool `Accounting` will be monitored every minute for percentage full. When the percentage of the SAVE devices are 80 percent full, the associated action script *SaveScript* will be executed each time the threshold of 80 percent is met.

In addition to the `monitor` command, the following commands can also be useful for monitoring SAVE devices:

`symsnap list -svp` *`PoolName`*

Displays the copy sessions related to the specified SAVE device pool.

`symsnap list -svp` *`PoolName`* `-savedev`

Lists the SAVE devices that are in the specified SAVE device pool.

`symsnap list -pools [-v]`

Lists all the SAVE device pools. The `-v` option displays a verbose listing for each pool.

`symsnap show pool` *`PoolName`*

Shows detailed information about the specified SAVE device pool.

Pairing an additional target device with each source device in a group

When working with either a composite or device group, you can use the `-concurrent` option with the `create` action to pair an additional target device with each source device in a group.

For example, to pair an additional target device with each source device in group `ProdDB`, enter:

```
symsnap -g ProdDB create -concurrent
```

When the copy session is created, an additional target device will be paired with each source device in the group. For example, if there were two target devices paired with each source device in the group before activating the session, there will be three target devices paired with each source device after the session is activated.

To verify that each source device in the group has multiple targets, enter:

```
symsnap -g ProdDB verify -created -concurrent
```

Activating a virtual copy session

To create a point-in-time image, you must activate the create copy session. To activate the copy session created in the example on page 76, use the following symsnap command:

```
symsnap -g ProdDB activate DEV001 vdev ld VDEV005
```

This activates the copy operation from the source device to the virtual target device. Activating the copy session starts the copy on first write mechanism and places the target device in the Read/Write state. The target host can access the copy and has access to data on the source host until the copy session is terminated.

Note: Virtual data is made available as a point-in-time copy at the time of activation and not at the time that the session was created.

Making the target device not ready to the host

The Not Ready (-not_ready) option can be used with the activate action to start the copy-on-first-write mechanism but causes the target device to remain not ready to its host, as follows:

```
symsnap -g ProdDB activate DEV001 vdev ld VDEV005 -not_ready
```

The copy session will be activated and the target device will be placed in the Not Ready state. The snap copy can later be Read/Write enabled to the host using the symld ready command.

Activating copy sessions consistently

You can consistently activate multiple virtual copy sessions involving a database using either PowerPath-connected devices or the Enginuity Consistency Assist (ECA) feature. These features allow snap copy sessions to be activated with a consistent, restartable copy of the database.

Using PowerPath

To activate a virtual copy session using the PowerPath (-ppath) option, you need to specify either the source device(s) (SRCDEVS) or the PowerPath device and any pre-action and post-action scripts.

The following example activates a virtual copy session for the device group ProdDB using PowerPath source devices and scripts:

```
symsnap -g ProdDB activate -ppath SRCDEVS -preaction StartScript
-postaction EndScript
```

Using ECA

You can use the Enginuity Consistency Assist (ECA) feature to activate virtual copy sessions that are consistent with the database up to the point in time that the activation occurs. The feature suspends writes to the source device during the activation.

Use the symsnap activate command with the consistent (-consistent) option to invoke ECA. When the activation has completed, writes are resumed and the target device contains pointers for a consistent production database copy of the source device at the time of activation.

To consistently activate copy sessions using ECA, you must have either a control host with no database or a database host with a dedicated channel. This will ensure that in write intensive environments SYMAPI will be able to activate the copy sessions within the ECA window, regardless of the number of outstanding I/Os held by the HBA. Refer to Figure 16, for a depiction of how a control host can consistently activate a copy session involving three database hosts that access devices on a Symmetrix array.

A Symmetrix device group, composite group, or a device file must be created on the controlling host for the target database to be consistently activated. Device groups can be created to include all of the devices being accessed or defined by database host access. For example, if you define a device group that includes all of the devices being accessed by Hosts A, B, and C (Figure 16), then you can consistently activate all of the copy sessions related to those hosts with a single command. However, if you define a device group that includes only the devices accessed by Host A, then you can activate those copy sessions related to Host A without affecting the other hosts.

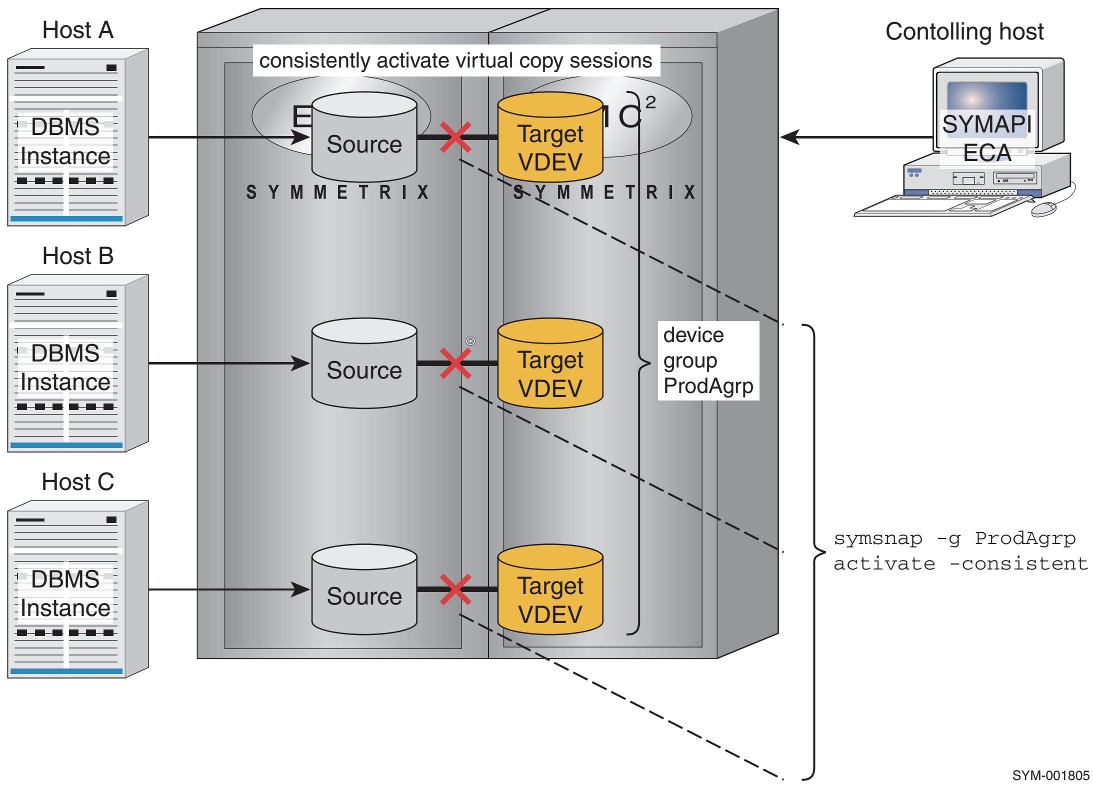

Figure 16 TF/Snap consistent activate using ECA

Activating an additional copy session for each device pair in a group

When working with either a composite or device group, you can use the `-concurrent` option with the `activate` action to activate an additional copy session for each device pair in a group.

For example, to activate an additional copy session for each device pair in group `ProdDB`, enter:

```
symsnap -g ProdDB activate -concurrent
```

Restoring data from virtual devices

Three types of restore operations can be performed for virtual device copy sessions:

- Incremental restore back to the original source device.
- Incremental restore to a BCV, which has been split from its original standard source device but maintains the incremental relationship with the source.
- Full restore to any standard or split BCV device outside of the existing copy session. The target device of the restore must be of the same size and emulation type as the source device.

A new restore copy session between the source device and the restore target device is created. A restore operation can only be performed if an additional copy session is available for use. Refer to "Understanding copy session limits" on page 76 for an explanation.

By default, any existing copy sessions persist until manually terminated by using the `symsnap terminate` command. After the virtual device has been restored to another device, the restore copy session must be terminated first, before another restore operation is allowed from that virtual device. Use the `symsnap query -multi` command to view all existing targets paired with a source device. The original snap copy session displays as being in the CopyOnWrite state, and the restore copy session displays as being either in the RestInProg or Restored state.

Preserving existing snap copy sessions

Since Enginuity Version 5670, the persistent restore option is enabled by default. Persistent restore preserves all existing snap copy sessions even after the virtual device has been restored to another device. If you want to disable the persistent restore option so that the original snap copy session for the restore is automatically terminated, you can set the following parameter in the options file to `DISABLE`:

```
SYMAPI_SNAP_PERSISTENT_RESTORE = ENABLE | DISABLE
```

Note: Persistent restore cannot be disabled in Enginuity Version 5671 and higher.

Incrementally restoring to a source

The following command-line entry shows an incremental restore operation back to the original source device `DEV001` from the virtual device `VDEV005`:

```
symsnap restore DEV001 vdev ld VDEV005
```

In this example, through the use of device pointers to the SAVE device, the virtual device `VDEV005` will be incrementally restored back to `DEV001`. Any changes made to the virtual device tracks during the active copy session will be restored back to the original source device.

Note: Be aware that any changes written to the original source device during the copy session will be overwritten by the virtual device tracks when the source device is restored.

The restore target device (source device `DEV001`) and virtual device (`VDEV005`) are automatically set to the Not Ready state while the track protection bitmaps are set up to copy any changed tracks. `DEV001` automatically becomes available for use (Ready state) as soon as the track protection bitmap completes. The changed tracks then begin copying and will continue to copy in the background until all the protected tracks have been restored.

If you want to continue using the original copy session, the virtual device must be manually set to the Ready state after the restore operation has completed (all tracks are copied back to the source). Use one of the following commands to set the virtual device to the Ready state after the above restore operation:

```
symld -g Group1 ready -vdev VDEV005
symdev -sid SymmID ready 001C
```

Once the restore completes, both the original and restore copy sessions are maintained and must be terminated manually if not needed for future use. When a original session is terminated, the device pointers are deleted from the virtual device and the SAVE device space is freed for future use.

Note: The restored copy session must be terminated first, before the original copy session is allowed to be terminated.

To terminate the restored copy session, enter:

```
symsnap terminate DEV001 vdev ld VDEV005 -restored
```

To terminate the original copy session, enter:

```
symsnap terminate DEV001 vdev ld VDEV005
```

Incrementally restoring to a BCV

The following command-line entry shows an incremental restore operation to a split BCV device `BCV001` from the virtual device `VDEV005`:

```
symsnap restore BCV001 vdev ld VDEV005
```

To restore to a BCV device, the BCV device must be split from its standard device. The point-in-time snap copy gets restored to the BCV as shown in Figure 17 on page 84. However, any further changes made to the point-in-time snap copy (from the attached host to the VDEV), during the active copy session, will also be included in the restore action to the split BCV device (`BCV001`). A new copy session begins between the source and target BCV and any changed tracks pointed to by the pointers are then copied to the split BCV from the source and SAVE devices. (Only the tracks that are different between the BCV and VDEV are copied.) Once the restore completes, both the original and restore copy sessions are maintained until manually terminated.

Note: You cannot restore to a BCV running in emulation mode.

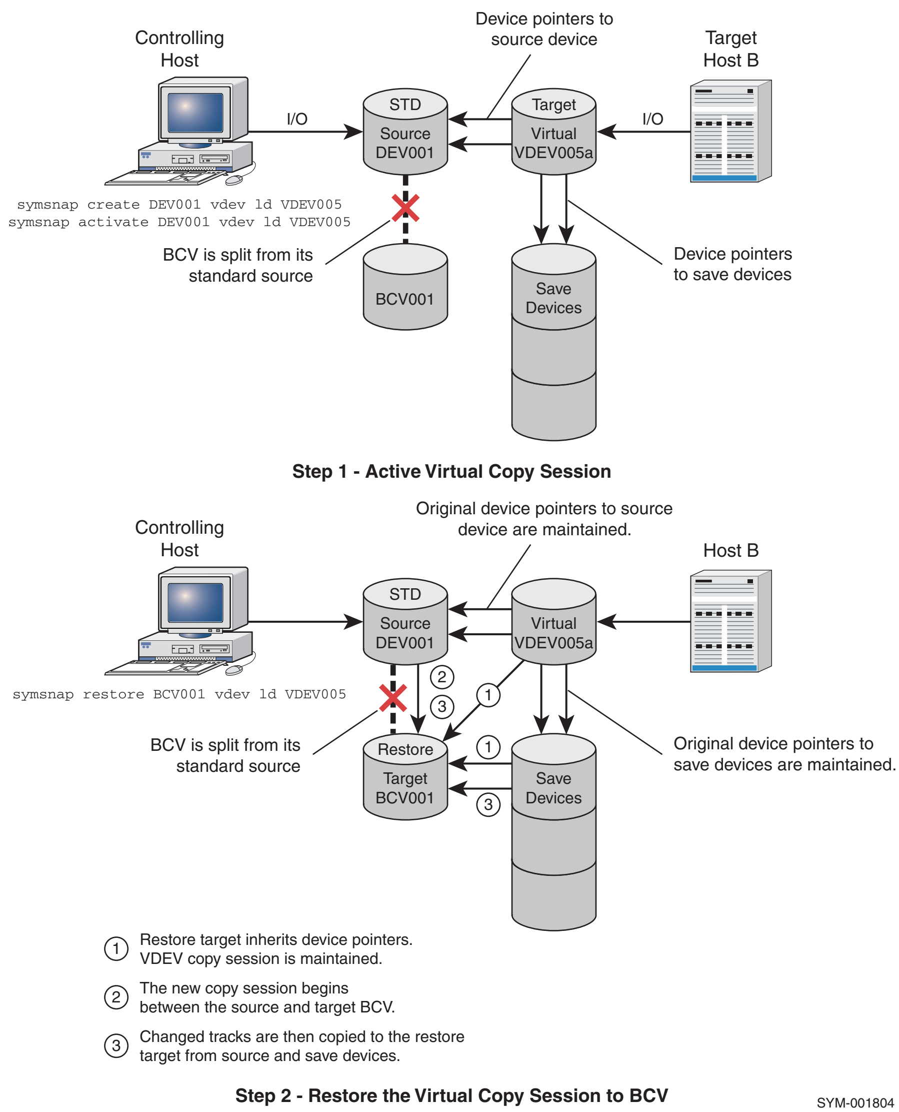

Figure 17 **Incremental restore to a BCV**

The target device (`BCV001`) and the virtual device (`VDEV005`) are automatically set to the Not Ready state while the track protection bitmaps are set up to copy any changed tracks. `BCV001` automatically becomes available for use (Ready state) as soon as the track protection bitmap completes. The changed tracks then begin copying and will continue to copy in the background until all protected tracks have been restored.

If you want to continue using the original copy session, the virtual device must be manually set to the Ready state after the restore operation has completed (all tracks are copied to the split BCV). Use one of the following commands to set the virtual device to the Ready state after the above restore operation:

```
symld -g Group1 ready -vdev VDEV005
symdev -sid SymmID ready 001D
```

Note: The restored copy session must be terminated first, before the original copy session is allowed to be terminated.

To terminate the restored copy session, enter:

```
symsnap terminate DEV001 bcv ld BCV001 -restored
```

To terminate the original copy session, enter:

```
symsnap terminate DEV001 vdev ld VDEV005
```

Optionally, after terminating the copy session, you can also issue a TimeFinder `symmir restore` operation from the restored BCV back to the standard source device. For information on reestablishing and restoring BCV pairs, refer to Chapter 4.

Fully restoring to anywhere

A full device restore operation from the virtual device is allowed to any device of the same size and emulation type as the source device. The following example uses the `restore` command with the `-full` option to fully restore a virtual device `VDEV005` to another standard device `DEV004`:

```
symsnap -full restore DEV004 vdev ld VDEV005
```

Any changes made to the virtual device (`VDEV005`) during the active copy session will be restored to the specified device (`DEV004`). The restore target inherits the virtual device pointers. A new copy session begins between the source and target device and any changed tracks pointed to by the pointers are then copied to the target device from the source and SAVE devices. Upon completion of the restore operation, both the original and restore copy sessions are maintained until manually terminated.

The target device (`DEV004`) and the virtual device (`VDEV005`) are automatically set to the Not Ready state while the track protection bitmaps are set up to copy any changed tracks. `DEV004` automatically becomes available for use (Ready state) as soon as the track protection bitmap completes. The changed tracks then begin copying and will continue to copy until all protected tracks have been restored.

If you want to continue using the original copy session, the virtual device must be manually set to the Ready state after the restore operation has completed (all tracks are copied to the restore target). Use one of the following commands to set the virtual device to the Ready state after the above restore operation:

```
symld -g Group1 ready -vdev VDEV005
symdev -sid SymmID ready 001E
```

When a session is terminated, all device pointers are deleted from the virtual device and the SAVE device space is freed for future use.

Note: The restored copy session must be terminated first, before the original copy session is allowed to be terminated.

To terminate the restored copy session, enter:

```
symsnap terminate DEV001 sym ld DEV004 -restored
```

To terminate the original copy session for a virtual device, enter:

```
symsnap terminate DEV001 vdev ld VDEV005
```

Recreating a virtual copy session

Starting with Enginuity 5874, snap sessions can be recreated on an existing virtual device (VDEV) in preparation to activate a new point-in-time image. Snap `recreate` is only valid when issued against sessions that have been previously activated. This process makes it more convenient to reuse a virtual device to acquire a new point-in-time image.

Since the `recreate` operation replaces the previous point-in-time image with a new one, the used tracks in the SAVE devices that were associated with the previous session are freed during the processing of this command.

Recreating a virtual snap copy session requires the following steps:

1. Create a snap session. (Refer to page 76.)
2. Activate a snap session. (Refer to page 80.)
3. Recreate a snap session.
4. Activate a snap session. (Refer to page 80.)
5. Repeat the Recreate/Activate a snap session, as necessary.
6. Terminate the snap session when no longer needed. (Refer to page 87.)

To recreate the copy session activated in the example on page 80 use the following `symsnap` command:

```
symsnap -g ProdDB recreate DEV001 vdev ld VDEV005
```

The Snap recreate functionality is supported with the `symsnap query` and `symsnap verify` commands. For example:

```
symsnap -g TestDg query -multi
symsnap -g TestDg verify -recreated
```

Note: You can use the `-recreated` parameter in combination with other verify states when verifying for multiple states. When multiple verify states are listed, the states are evaluated in a logical OR fashion so the result will be true if the session is in any of the listed states.

The following restrictions apply to the Snap Recreate functionality:

- The session to be recreated must be in a CopyOnWrite or Copied state.
- Any restore session that exists on the device must be terminated prior to issue the recreate operation.
- Snap Recreate is not supported for Multivirtual snap operations. This is a limitation in Enginuity release level 5874.

Terminating a virtual copy session

To terminate the copy session activated in the example on page 80 use the following `symsnap` command:

```
symsnap -g ProdDB terminate DEV001 vdev ld VDEV005
```

Terminating a copy session deletes the pairing information in the Symmetrix array and removes any hold on the target device. Terminating the session causes the target host to lose access to data pointed to by the virtual device.

Terminating a session while the device pairs are in the CopyOnWrite state will cause the session to end. Once the virtual copy session is terminated, the information is no longer available on the virtual device.

If a copy session has been restored, the restored session must be terminated first, before the original copy session is allowed to be terminated.

Note: If the state is RestInProg, then the `-symforce` option must be applied to terminate the session.

Using a BCV as the snap source

As shown in Figure 18 on page 89, you can create a virtual copy session between a BCV source device and a virtual target device. The controlling host performs I/O to the standard device that is established with a BCV as part of a BCV pair. At some point, when the BCV is synchronized with the standard device, you can split the BCV from the standard and create a point-in-time copy of the BCV. The split operation must be entirely complete including the background phase before you can create a copy session on it.

Chapter 4 provides additional information on using the `symmir` command and how to perform an instant split operation.

Pair states ruling snap operations

Because various other ongoing operations can conflict with your copy session, certain rules must be considered. The availability of some snap copy operations depends on the current SRDF state, clone state, and BCV pair state. The following rules apply to certain BCV pair states:

- If the source of a `symsnap create` or `activate` operation is a BCV, the BCV pair state must be Split. The Split must be totally complete before the operation is allowed.

  Note: The existing BCV snap copy session must be terminated before the BCV can be reestablished or restored again with its standard device.

- A TimeFinder standard source device cannot be created, activated or restored in a copy session if the BCV pair state is SplitBfrRest or RestInProg.

  Note: If you have an active snap copy session from a standard device that also has an established BCV and you attempt to restore the standard from the BCV, the restore operation may fail due to insufficient space on the SAVE device. In this case it may be best to first restore and terminate the snap copy session, then attempt the BCV restore operation.

- The `symsnap terminate` command is allowed for all BCV pair states.

For the TimeFinder pair states that rule TimeFinder/Snap copy sessions, refer to Appendix B.

For information regarding possible SRDF pair state conflicts, refer to "State rules for TimeFinder/Snap operations" on page 426.

For a description of each BCV pair state, refer to *BCV Pair States* in Chapter 4.

Example: Creating a virtual copy from a BCV

The following example creates a copy of source device `BCV001` in device group `ProdDB` on a Symmetrix array to target device `VDEV005` on the same array.

Once the copy session is activated, Host C can access the virtual target device tracks containing the device pointers to the point-in-time copy. If Host B writes to BCV device tracks, Symmetrix Enginuity software immediately copies the original tracks to the SAVE device before allowing new data to overwrite those BCV tracks.

Note: In this example, where multiple hosts have access to the BCV source, you should consider using the `-not_ready` option with the `split` command to make the BCV not ready. This will enable you to keep the same data on the BCV and virtual devices. If you decide to use this option, you may need to release any Not Ready state imposed on any devices once the session completes.

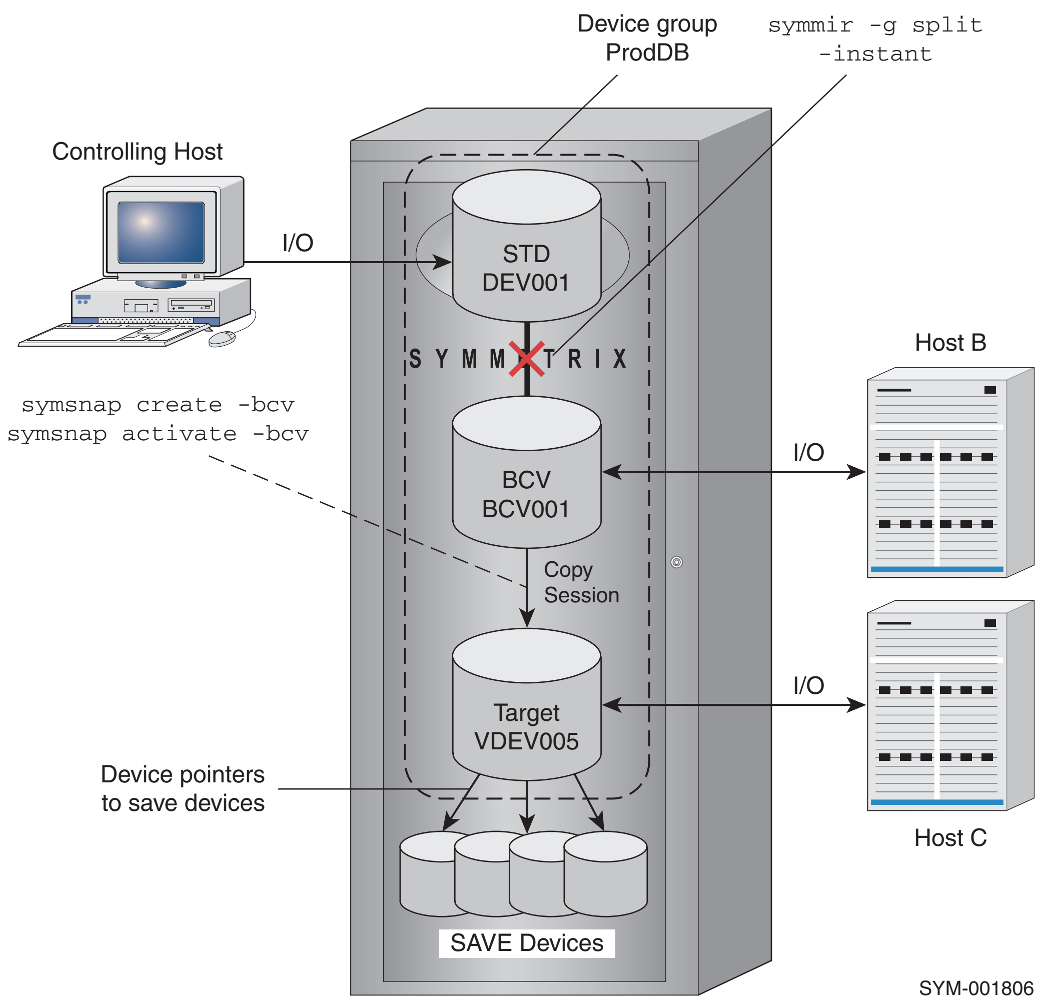

Figure 18 **Creating a virtual copy from a BCV**

For this example, the device pair was set in the Not Ready state.

The following steps outline the example shown in Figure 18 on page 89:

1. Perform an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it:

   ```
   symmir -g ProdDB split DEV001 -instant -not_ready -noprompt
   ```

2. Verify that the background split is complete. The following command checks every five seconds:

   ```
   symmir -g ProdDB verify DEV001 -split -bg -i 5
   ```

3. Create a copy session between the BCV source device `BCV001` and the virtual target device `VDEV005`*:*

   ```
   symsnap -g ProdDB create BCV001 vdev ld VDEV005
   ```

   or

   ```
   symsnap -g ProdDb create -bcv
   ```

   Note: Using the `-bcv` option applies the command within the device group to use BCV devices as the source and VDEV devices as the target.

4. Activate the copy session to Host C:

   ```
   symsnap -g ProdDB activate BCV001 vdev ld VDEV005
   ```

   or

   ```
   symsnap -g ProdDb activate -bcv
   ```

5. Query the state of the copy operation and verify the CopyOnWrite state:

   ```
   symsnap -g ProdDB query
   symsnap -g ProdDB verify BCV001 -copyonwrite
   ```

6. When the application has finished accessing the data, the copy session and pair relationship can be terminated:

   ```
   symsnap -g ProdDB terminate BCV001 vdev ld VDEV005 -noprompt
   ```

   or

   ```
   symsnap -g ProdDb terminate -bcv
   ```

7. Incrementally re-establish the BCV pair:

   ```
   symmir -g ProdDB establish DEV001 -noprompt
   ```

RDF2 Async device as Snap source device

Starting with Enginuity 5874, an RDF2 asynchronous device can be used as a Snap source device. This feature enables an RDF relationship to be put into asynchronous mode when the R2 device is a Snap source. This feature works with all modes of operation: file, DG, CG, LDEV, and ListDevList.

Creating multiple virtual copies

Since a total of 16 sessions can exist, you can create up to 15 virtual copy sessions and one reserved session (for restore operations) of a source device to various virtual target devices. A source device can concurrently copy data to up to 15 target devices at one time, depending on the available sessions.

To copy to multiple target devices, each target requires a separate copy session to become a host-addressable copy. Target host devices cannot access the virtual copy until the copy session is activated. Up to 15 copy sessions can be activated at the same time on the same source device. "Understanding copy session limits" on page 76 provides greater detail about the number of virtual copy sessions.

Note: When using multiple virtual copy sessions, you cannot use the Snap Recreate functionality. This is a limitation in Enginuity 5874.

Creating multiple virtual copies from a standard device

Figure 19 illustrates creating copy sessions for multiple targets from a standard source device `DEV001` to four virtual target devices (`VDEV005`, `VDEV006`, `VDEV007` and `VDEV008`) with various hosts accessing them.

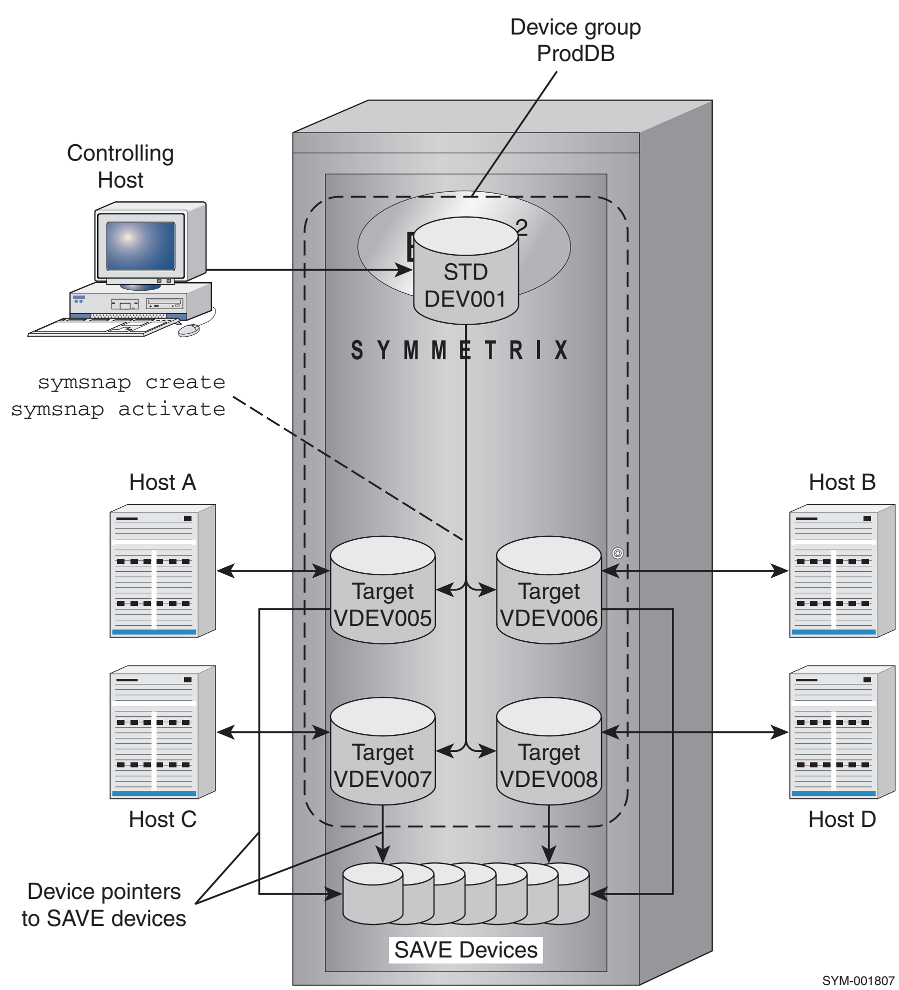

Figure 19 Creating multiple virtual copies from a standard device

---

Note: A separate copy session must be created between the source device (DEV001) and each target device (`VDEV005`, `VDEV006`, `VDEV007`, and `VDEV008`).

The following steps outline the example shown in Figure 19 on page 91:

1. Create a copy session between the standard source device `DEV001` and each of the four virtual target devices `VDEV005`, `VDEV006`, `VDEV007`, and `VDEV008`:

```
symsnap -g ProdDB create DEV001 vdev ld VDEV005
symsnap -g ProdDB create DEV001 vdev ld VDEV006
symsnap -g ProdDB create DEV001 vdev ld VDEV007
symsnap -g ProdDB create DEV001 vdev ld VDEV008
```

2. Activate the copy operation with one command to activate all sessions simultaneously:

```
symsnap -g ProdDB activate  DEV001 vdev ld VDEV005
                            DEV001 vdev ld VDEV006
                            DEV001 vdev ld VDEV007
                            DEV001 vdev ld VDEV008
```

3. Using the `query` argument, you can display the state of all devices involved in the copy operation by including the `-multi` option:

```
symsnap -g ProdDB query -multi
```

4. When the host devices have finished using the copy sessions, the pair relationships can be terminated individually as needed:

```
symsnap -g ProdDB terminate DEV001 vdev ld VDEV005 -noprompt
symsnap -g ProdDB terminate DEV001 vdev ld VDEV006 -noprompt
symsnap -g ProdDB terminate DEV001 vdev ld VDEV007 -noprompt
symsnap -g ProdDB terminate DEV001 vdev ld VDEV008 -noprompt
```

Creating multiple virtual copies from a BCV device

Figure 20 illustrates how to create multiple copy sessions from a BCV source device `BCV001` to four virtual target devices (`VDEV005`, `VDEV006`, `VDEV007`, and `VDEV008`) with various hosts accessing them.

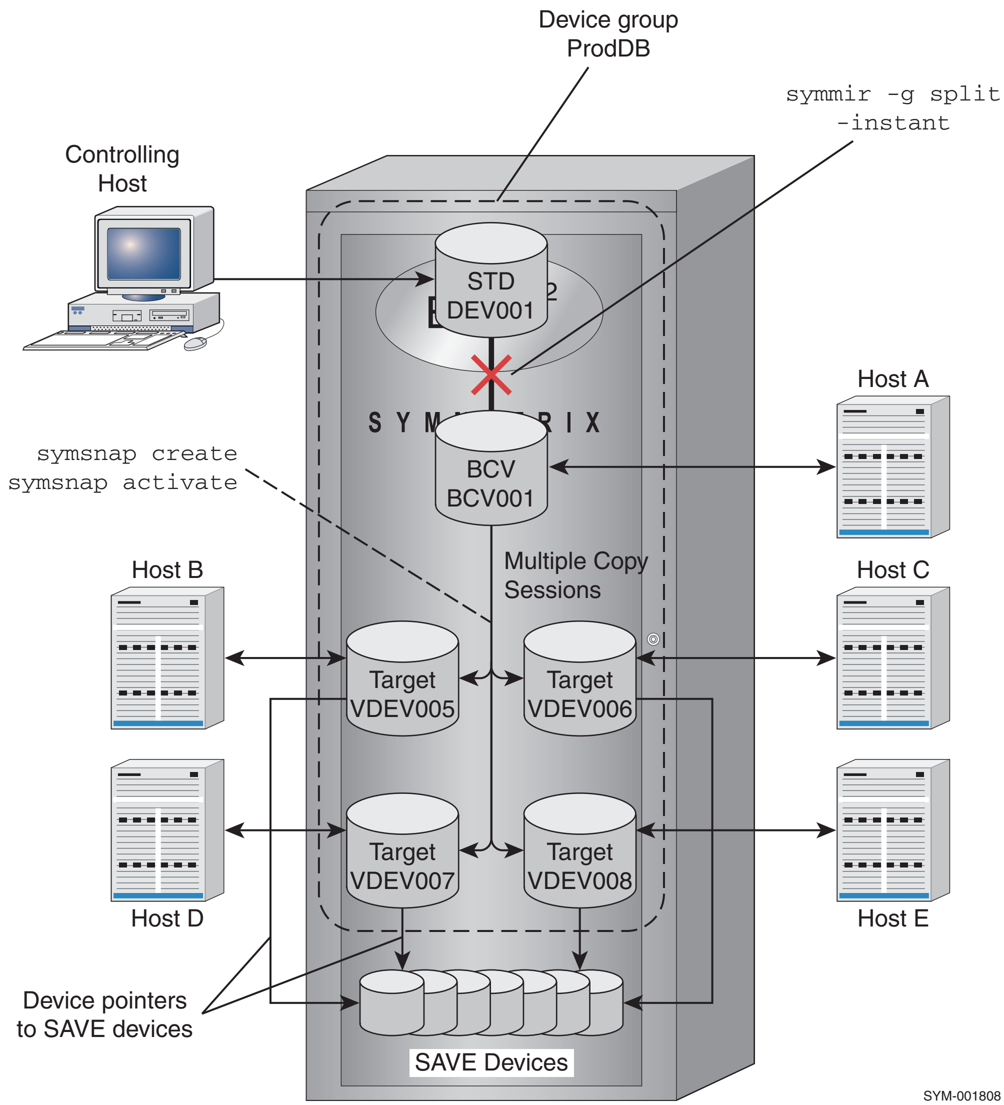

Figure 20 Creating multiple virtual copies from a BCV device

The following steps outline the example shown in Figure 20:

1. Perform an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to the device:

   `symmir -g ProdDB split DEV001 -instant -not_ready -noprompt`

   Note: Chapter 4 provides additional information on using the `symmir` command and performing an instant split operation.

2. Verify that the background split is complete. The following command checks every five seconds:

   `symmir -g ProdDB verify DEV001 -split -bg -i 5`

3. Create a copy session between the BCV source device `BCV001` and each of the four virtual target devices `VDEV005`, `VDEV006`, `VDEV007`, and `VDEV008`:

```
symsnap -g ProdDB create BCV001 vdev ld VDEV005
symsnap -g ProdDB create BCV001 vdev ld VDEV006
symsnap -g ProdDB create BCV001 vdev ld VDEV007
symsnap -g ProdDB create BCV001 vdev ld VDEV008
```

4. Activate the copy operation with one command to activate all sessions simultaneously:

```
symsnap -g ProdDB activateBCV001 vdev ld VDEV005
                          BCV001 vdev ld VDEV006
                          BCV001 vdev ld VDEV007
                          BCV001 vdev ld VDEV008
```

5. Using the `query` argument, you can display the state of all devices involved in the copy operation by including the `-multi` option:

```
symsnap -g ProdDB query -multi -BCV
```

6. When the host devices have finished accessing the data, the copy sessions and pair relationships can be terminated individually as needed:

```
symsnap -g ProdDB terminate BCV001 vdev ld VDEV005 -noprompt
symsnap -g ProdDB terminate BCV001 vdev ld VDEV006 -noprompt
symsnap -g ProdDB terminate BCV001 vdev ld VDEV007 -noprompt
symsnap -g ProdDB terminate BCV001 vdev ld VDEV008 -noprompt
```

7. Incrementally reestablish the BCV pair (`DEV001` and `BCV001`):

```
symmir -g ProdDB establish DEV001 -noprompt
```

Attaching source and target virtual devices

You can use the `symsnap attach` and `detach` arguments to set up preferred device pairs. Pre-determining attached device pairs eliminates the need to specify copy session target devices from within the device group for the `create` and `activate` arguments. The attached pairs will be used whenever a `symsnap` operation is requested for the specified device group. If a `symsnap create` or `activate` command does not specify a device pair from within the device group, the attached pair will automatically be used for the operation.

Note: The `attach` option can only be used when attaching a standard source device with a target VDEV device. You cannot use `symsnap attach` if the source device is a BCV device.

To set up a preferred attached device pair between source device `DEV001` in device group `ProdDB` and target device `VDEV002`, enter:

```
symsnap -g ProdDB attach DEV001 vdev ld VDEV002
```

To invoke a copy session operation from within a specified device group `ProdDB` using the pre-determined device pairs, enter:

```
symsnap -g ProdDB create
symsnap -g ProdDB activate
```

To detach (undo) the preferred device pair relationship, enter:

```
symsnap -g ProdDB detach DEV001 vdev ld VDEV002
```

Using composite groups to manage snap pairs across Symmetrix arrays

Figure 21 illustrates a production host that is locally connected to two Symmetrix arrays (A and B). A composite group is defined on the production host and includes source devices and target devices from these Symmetrix arrays. The target devices are virtual devices.

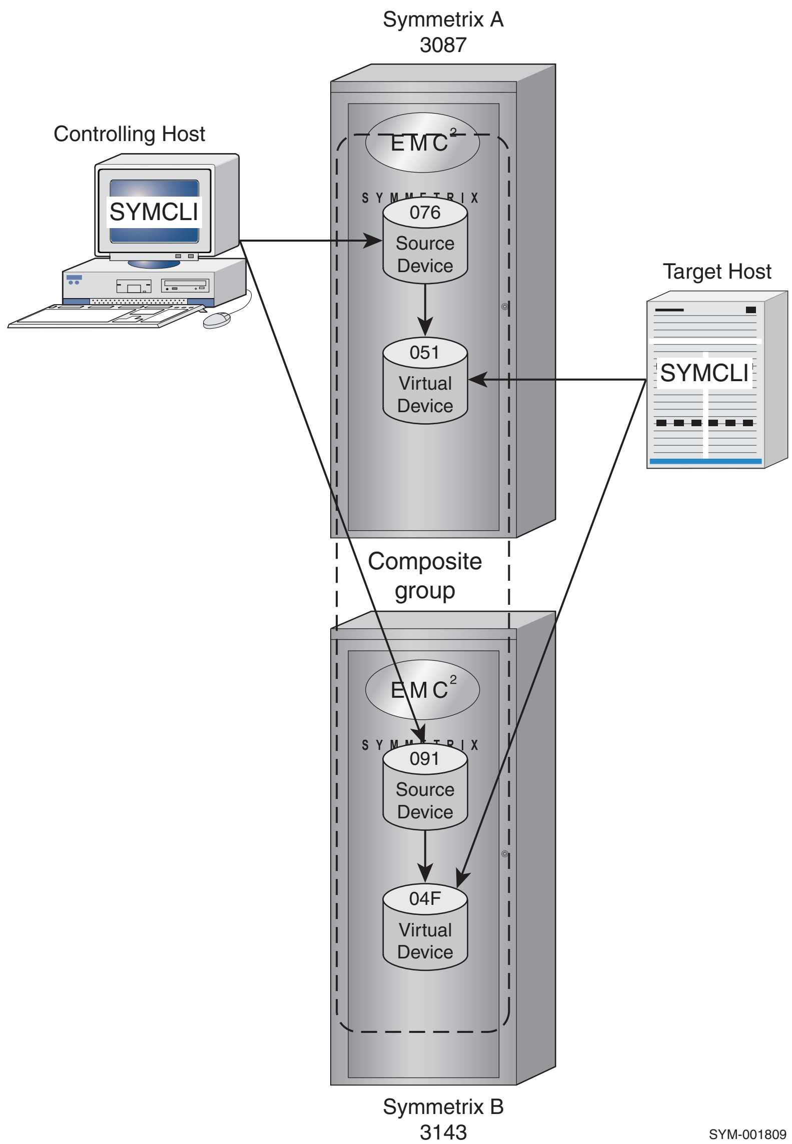

Figure 21 Using a composite group when a set of devices spans two Symmetrix arrays

Although snap operations might normally be performed from the production host because the composite group is defined there in its SYMAPI database, there are methods that would allow you to initiate copy sessions from another locally connected host like the target host. One way is to copy the composite group definition to another host. A more efficient method is to enable Group Name Services (GNS), which automatically propagates the composite group definition to the Symmetrix arrays and other locally attached hosts that are running the GNS daemon.

The following steps outline the setup required for controlling a set of snap pairs that spans two Symmetrix arrays as shown in Figure 21 on page 96:

1. From the production host, create a Regular type composite group (for example, `MyGrp`):

   ```
   symcg create MyGrp -type regular
   ```

2. Add to the composite group those standard devices on Symmetrix A (3087) and Symmetrix B (3143) that are the source devices:

   ```
   symcg -cg MyGrp -sid 3087 add dev 0076
   symcg -cg MyGrp -sid 3143 add dev 0091
   ```

3. Add a virtual device from each Symmetrix array to the composite group:

   ```
   symcg -cg MyGrp -sid 3087 add dev 0051 -vdev
   symcg -cg MyGrp -sid 3143 add dev 004F -vdev
   ```

4. Create snap pair sessions from those devices in the composite group:

   ```
   symsnap -cg MyGrp create
   ```

5. Activate the snap pair sessions:

   ```
   symsnap -cg MyGrp activate
   ```

Composite group support allows the source of a snap operation to be a standard device (as shown in step 2 above) or a BCV device. If you add BCV devices to the composite group for the purpose of being snap sources, you must use the `-bcv` option with the `symsnap` commands. For example, the following command creates snap pairs in the composite group using the BCVs as source devices:

```
symsnap -cg MyGrp create -bcv
```

You can add non-source BCV devices to the composite group for the purpose of restore operations. Composite group support allows you to restore to either the original source standard device or to a BCV that has been split from its paired standard device. The following commands split all BCV pairs in the composite group and perform a restore operation from the virtual devices to the BCVs:

```
symmir -cg MyGrp split
symsnap -cg MyGrp restore -bcv
```

For more information about performing an incremental restore to a BCV that is split from the original source device, refer to "Incrementally restoring to a BCV" on page 83.

You can control specific snap pairs within the composite group. For example, to create and activate the DEV001/VDEV002 snap pair from devices in the group:

```
symsnap -cg MyGrp create DEV001 vdev ld VDEV002
symsnap -cg MyGrp activate DEV001 vdev ld VDEV002
```

Snapping a copy on a remote Symmetrix array

You can use a device group or composite group to snap devices on a remote Symmetrix array as shown in Figure 22. Performing SYMCLI commands from the controlling host allows the remote virtual device to receive a copy of the data from the R2 device. Remote host A can access the snap data.

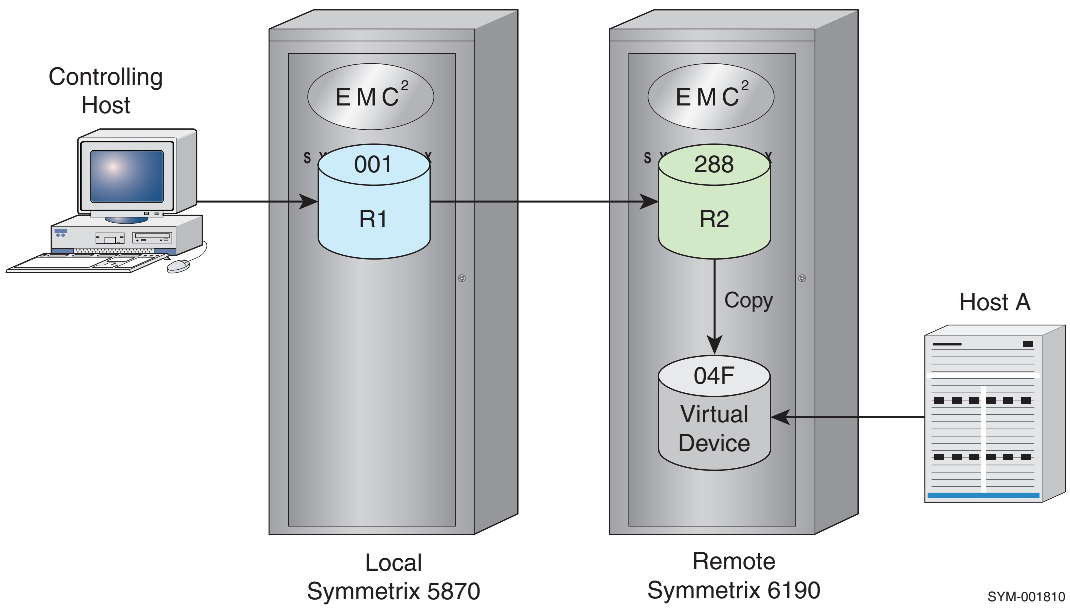

Figure 22 Snapping a copy on a remote Symmetrix array

The following steps outline an example of snapping local device data on a remote Symmetrix array:

1. Create an RDF1-type device group or composite group (for example, a device group named `remotesnaps`):

   ```
   symdg create remotesnaps -type rdf1
   ```

2. Add to the device group an R1 standard device (`001`) to hold the production data. Add a virtual device (`04F`) on the remote Symmetrix to hold the snap copy:

   ```
   symld -g remotesnaps add dev 001 -sid 5870
   symld -g remotesnaps add dev 04F -rdf -vdev
   ```

   If the local Symmetrix array has more than one RDF group (that is, concurrent RDF) linking the remote Symmetrix, include the RDF group option (for example, `-rdfg 2`) when adding the remote virtual device.

3. Create a snap pair session from the devices in the device group:

   ```
   symsnap -g remotesnaps create -rdf DEV001 vdev ld RVDEV001
   ```

4. Activate the snap pair session:

   ```
   symsnap -g remotesnaps activate -rdf DEV001 vdev ld RVDEV001
   ```

5. To query the progress of the remote snap operation:

   ```
   symsnap -g remotesnaps query -rdf
   ```

Snapping a copy from a remote BCV

Using SRDF technology and the TimeFinder RDF flag (the `-rdf` option), you can snap a copy from a BCV located on a remote Symmetrix array (Figure 23). By doing this, the remotely associated BCV is synchronized with the R2 device until you decide to split the BCV pair and snap a copy from the BCV to the virtual device.

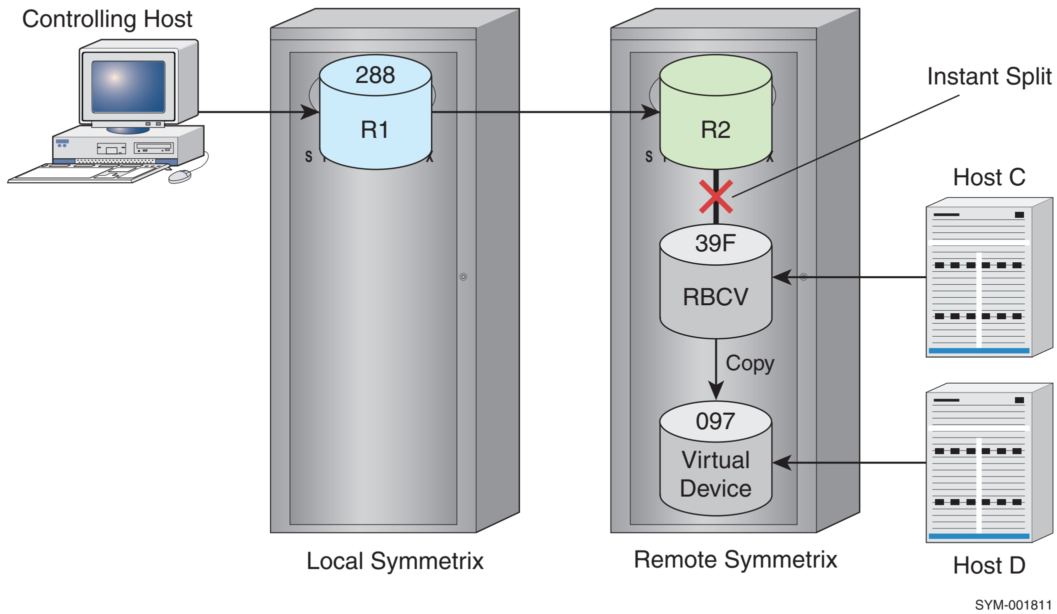

Figure 23 Snapping from a remote BCV source device

The following steps outline an example of creating a copy session between a remote BCV source device and a remote virtual device:

1. Create an RDF1-type device group or composite group (for example, a device group named `Rdf1Grp`):

   ```
   symdg create Rdf1Grp -type rdf1
   ```

2. Add to the device group an R1 standard device on the local Symmetrix array to hold production data. Add a virtual device on the remote Symmetrix array to hold the snap copy:

   ```
   symld -g Rdf1Grp add dev 288
   symld -g Rdf1Grp add dev 097 -rdf -vdev
   ```

3. Associate with the device group an RBCV to be the snap source on the remote Symmetrix array:

   ```
   symbcv -g Rdf1Grp associate dev 39F -rdf
   ```

4. Establish standard device 288 (`DEV001`) with the RBCV:

   ```
   symmir -g Rdf1Grp establish -full DEV001 -rdf
   ```

5. When the remote BCV pair is fully synchronized, perform an instant split on the BCV pair:

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

6. When the background split is complete, create and activate a copy session between the remote BCV source device and the remote virtual device:

```
symsnap -g Rdf1Grp snap create -rbcv RBCV001 vdev ld RVDEV001
symsnap -g Rdf1Grp snap activate -rbcv RBCV001 vdev ld RVDEV001
```

7. To query the progress of the remote snap operation or verify the copy session, you can issue the following commands that examine the snapped pair (the RBCV source device and the virtual device):

```
symsnap -g Rdf1Grp query -rbcv
symsnap -g Rdf1Grp verify -rbcv
```

Snapping copies of a source device's data locally and remotely

Figure 24 combines elements of Figures 22 and 23 to illustrate how you can snap copies of a source device's data to devices on two Symmetrix arrays. By splitting the BCV pairs on both sides at the same time with a consistent split operation, the BCVs on both sides contain data that is consistent with the R1 source data up to the time of the split. The controlling host can then perform a local snap and a remote snap, resulting in the virtual devices on both sides having copies of the source device's data (provided that the BCVs were not written to by their hosts prior to the snap operation).

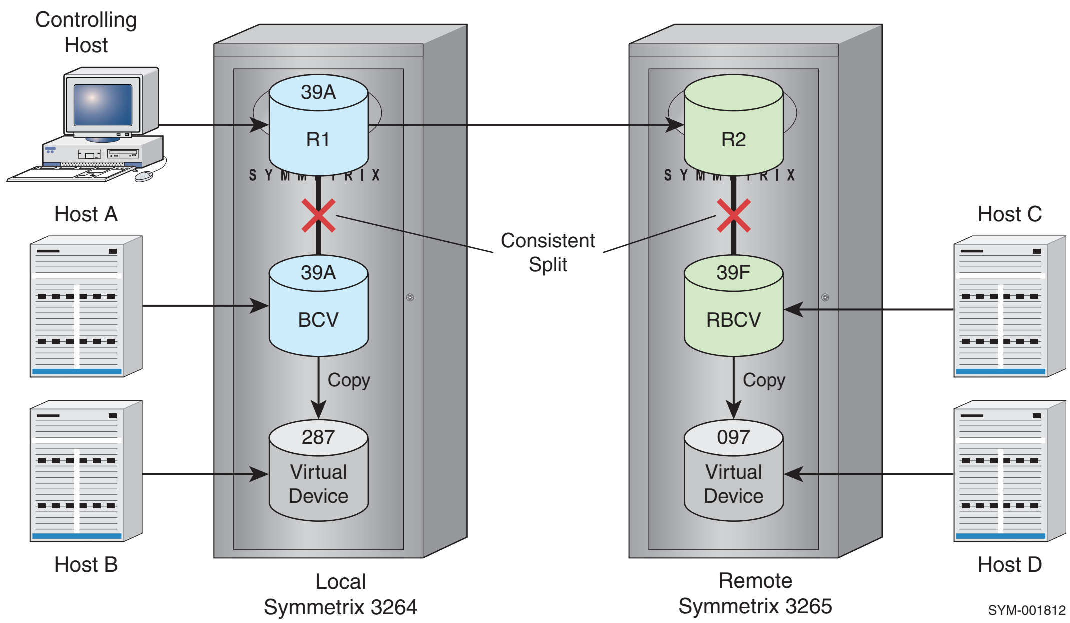

Figure 24 Snapping copies on local and remote Symmetrix arrays

The following steps outline the example shown in Figure 24:

1. Create an RDF1-type device group or composite group (for example, a device group named `Rdf1Grp`):

```
symdg create Rdf1Grp -type rdf1
```

2. Add to the device group an R1 standard device (`286`) on the local Symmetrix array (`sid 3264`) to hold production data. Add local virtual device `287` and remote virtual device `097` to hold the snap copies:

```
symld -g Rdf1Grp -sid 3264 add dev 286
symld -g Rdf1Grp -sid 3264 add dev 287 -vdev
symld -g Rdf1Grp -sid 3264 add dev 097 -rdf -vdev
```

3. Associate with the device group local BCV `39A` and remote BCV `39F`:

```
symbcv -g Rdf1Grp -sid 3264 associate dev 39A
symbcv -g Rdf1Grp -sid 3264 associate dev 39F -rdf
```

4. Establish the BCV pairs on both Symmetrix arrays. DEV001 is the logical device name of the R1 source device (`286`):

```
symmir -g Rdf1Grp establish -full DEV001
symmir -g Rdf1Grp establish -full DEV001 -rdf
```

5. When the BCV pairs are fully synchronized, perform a consistent split on both BCV pairs. This operation can be done with one command, using the `-both_sides` option. For example:

   ```
   symmir -g Rdf1Grp split -consistent -both_sides
   ```

   Note: To use the `-consistent` and `-both_sides` options, the SRDF pairs must be synchronized and in SRDF mode SYNCHRONOUS.

   ---

6. When the background split is complete, create the local and remote snap pairs:

   ```
   symsnap -g Rdf1Grp create -bcv BCV001 vdev ld VDEV001
   symsnap -g Rdf1Grp create -rbcv RBCV002 vdev ld RVDEV002
   ```

7. Activate the copy sessions for the local and remote snap pairs:

   ```
   symsnap -g Rdf1Grp activate -bcv BCV001 vdev ld VDEV001
   symsnap -g Rdf1Grp activate -rbcv RBCV002 vdev ld RVDEV002
   ```

8. To query the progress of the local snap operation or verify when the local copy is complete, you can issue the following commands that examine the local snap pair:

   ```
   symsnap -g Rdf1Grp query -bcv
   symsnap -g Rdf1Grp verify -bcv
   ```

9. To query the progress of the remote snap operation or verify when the remote copy is complete, you can issue the following commands that examine the remote snap pair:

   ```
   symsnap -g Rdf1Grp query -rbcv
   symsnap -g Rdf1Grp verify -rbcv
   ```

10. When subsequent snaps are no longer required, you can terminate the copy sessions by issuing commands that end copy sessions for the local and remote snap pairs:

    ```
    symsnap -g Rdf1Grp terminate -bcv BCV001 vdev ld VDEV001 -bcv
    symsnap -g Rdf1Grp terminate -rbcv RBCV002 vdev ld RVDEV002
    ```

Snapping multiple copies

The configuration in Figure 25 is basically the same as Figure 24 on page 101 except that this configuration snaps from a BCV source to four virtual devices on each Symmetrix array instead of to one virtual device on each Symmetrix array. This illustrates how a single `symsnap` command can create an image of a source device (a BCV in this case) on four virtual devices simultaneously.

As mentioned previously, by splitting the BCV pairs on both sides of an SRDF configuration at the same time, the BCVs on both sides contain data that is consistent with the R1 source data up to the time of the split. The controlling host can then perform a local snap and a remote snap, resulting in virtual devices on both sides having copies of the source device's data (provided that the BCVs were not written to by their hosts prior to the snap operation). In this configuration, 10 hosts can have access to copies of the source device's date (two copies on BCVs, and eight copies on virtual devices).

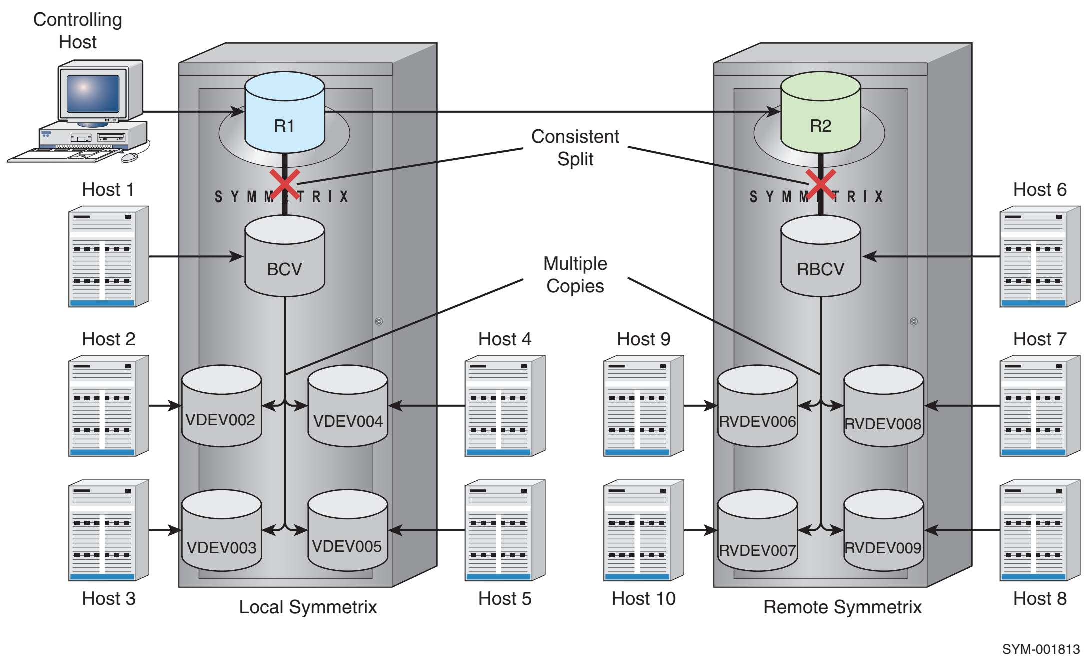

Figure 25 Snapping multiple copies on local and remote Symmetrix arrays

The process for building an RDF1-type device group is similar to the previous section. The difference is that a device group for this configuration contains the R1 standard device, two BCVs, and eight virtual devices. Establishing and splitting both BCV pairs is the same as the previous section.

What is different about this configuration is the ability to snap copies from a single source to multiple virtual devices simultaneously with a single command. Before initiating the snap operation, however, you must wait for completion of the BCV-pair background split involving the BCV that will be the source for the snap.

To snap copies to the four local virtual devices (`DEV002` – `DEV005`) from the local BCV source device (`BCV001`), issue the `symsnap create` and the `symsnap activate` commands:

```
symsnap -g Rdf1Grp create -bcv BCV001 vdev ld VDEV002
symsnap -g Rdf1Grp create -bcv BCV001 vdev ld VDEV003
symsnap -g Rdf1Grp create -bcv BCV001 vdev ld VDEV004
symsnap -g Rdf1Grp create -bcv BCV001 vdev ld VDEV005


symsnap -g Rdf1Grp activate -bcv BCV001 vdev ld VDEV002
                                 BCV001 vdev ld VDEV003
                                 BCV001 vdev ld VDEV004
                                 BCV001 vdev ld VDEV005
```

To snap copies to the four remote virtual devices (`DEV006` – `DEV009`) from the remotely associated BCV source device (BCV002), issue the `symsnap` commands with the `-rbcv` option:

```
symsnap -g Rdf1Grp create -rbcv RBCV002 vdev ld RVDEV006
symsnap -g Rdf1Grp create -rbcv RBCV002 vdev ld RVDEV007
symsnap -g Rdf1Grp create -rbcv RBCV002 vdev ld RVDEV008
symsnap -g Rdf1Grp create -rbcv RBCV002 vdev ld RVDEV009


symsnap -g Rdf1Grp activate -rbcv RBCV002 vdev ld RVDEV006
                                  RBCV002 vdev ld RVDEV007
                                  RBCV002 vdev ld RVDEV008
                                  RBCV002 vdev ld RVDEV009
```

Snapping a copy at the tertiary site of a Cascaded SRDF configuration

Using SRDF technology and the TimeFinder hop 2 flag (`-hop2` option), you can snap devices on a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration (Figure 26). Performing SYMCLI commands from the controlling host allows the remote virtual device to receive a copy of the data from the R1 device. Remote host A can access the snap data.

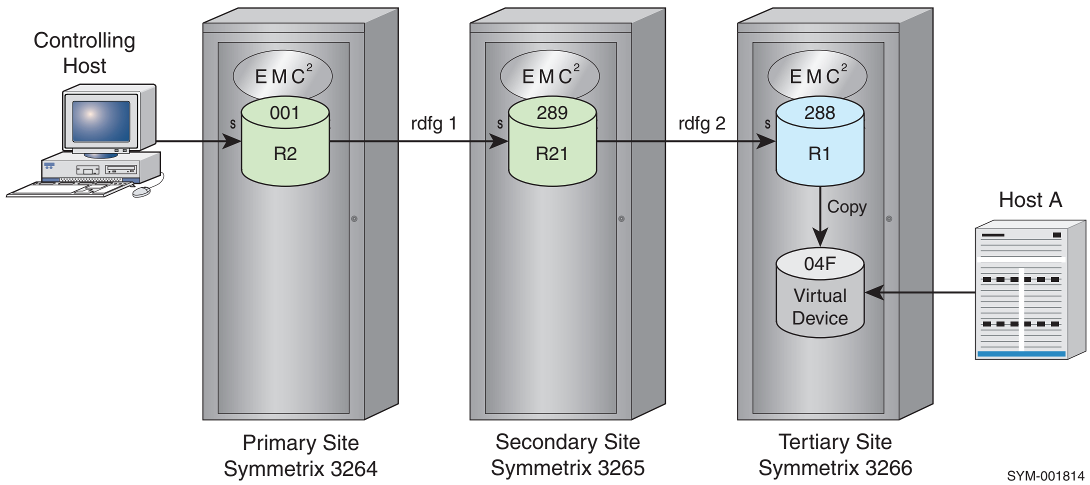

Figure 26 Snapping a copy at the tertiary site of a Cascaded SRDF configuration

The following steps outline an example of snapping local device data on a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration:

1. Create an RDF2-type device group or composite group (for example, a group named `remotesnaps`):

   To create an RDF2-type device group:

   ```
   symdg create remotesnaps -type rdf2
   ```

   To create an RDF2-type composite group:

   ```
   symcg create remotesnaps -type rdf2
   ```

2. Add devices to the group. From the Symmetrix array at the primary site, add an R2 standard device. From the Symmetrix array at the tertiary site, add a virtual device to hold the snap copy.

   To add devices to a device group:

   ```
   symld -g remotesnaps -sid 3264 add dev 001
   symld -g remotesnaps -hop2 -rdfg 1 -remote_rdfg 3 -vdev add dev 04f
   ```

   To add devices to a composite group:

   ```
   symcg -cg remotesnaps -sid 3264 add dev 001

   symcg -cg remotesnaps -sid 3264 -hop2 -rdfg 1 -remote_rdfg 3 -vdev
   add dev 04f
   ```

3. Create a snap pair session from the devices in the group:

   ```
   symsnap -g remotesnaps -exact -hop2 create
   ```

4. Activate the snap pair session:

   symsnap -g remotesnaps -hop2 activate

5. To query the progress of the remote snap session:

   symsnap -g remotesnaps query -hop2

6. To terminate the remote snap session:

   symsnap -g remotesnaps -hop2 terminate

Snapping a copy from a clone target device

You can perform snap operations from clone target devices. This feature allows the target device of a clone session to be used as a source device for one or more snap sessions.

The following restrictions apply to snap from clone target devices:

- The original clone session (`symclone`) must be in the copied state.
- After one or more snap sessions begin using the original clone devices, the only action permitted on the original clone session is `terminate`. All other actions are blocked until all of the snaps are terminated.

  The only actions permitted on a snap session with a clone session target as their source are activate and terminate. All other actions are blocked until the original clone session is terminated.

For example:

```
symclone -g DgName create -copy -dff
symclone -g DgName activate
symsnap -g DgName -bcv create
symsnap -g DgName -bcv activate
symsnap -g DgName -bcv terminate
symclone -g DgName recreate
```

Command options with device groups or composite groups

Options to the `symsnap -g` and `-cg` command line arguments provide more action flexibility to control copy sessions when you are operating on device(s) of a specified device group or composite group. Table 8 lists the `symsnap` control operations and the possible options to use when targeting a specified device group or composite group.

Table 8 symsnap -g and -cg control arguments and possible options (page 1 of 2)

	Argument action								
Options	**create**	**activate**	**recreate**	**terminate**	**restore**	**query**	**verify**	**attach**	**detach**
`-bcv`	✔	✔	✔	✔	✔	✔	✔		
`-both_sides`		✔							
`-c`, `-i`	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-concurrent`	✔	✔	✔				✔		
`-consistent`		✔							
`-copied`							✔		
`-copyonwrite`							✔		
`-created`							✔		
`-exact`	✔								
`-force`	✔	✔	❖	✔	✔				
`-full`					✔				
`-hop2`	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-multi`						✔			
`-noprompt`	✔	✔	✔	✔	✔			✔	✔
`-not_ready`		✔			✔				
`-offline`						✔	✔		
`-ppath`		✔							
`-preaction`, `-postaction`		✔							
`-preservetgtlocks`[a], `-lockid`[a]	✔	✔	✔	✔	✔				
`-rbcv`	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-rdb`, `-db`, `-dbtype`		✔							
`-rdf`	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-restinprog`							✔		
`-restore`						✔			
`-restored`				✔			✔		
`-skip`	✔	✔	✔	✔					

Table 8 symsnap -g and -cg control arguments and possible options (page 2 of 2)

	Argument action								
Options	**create**	**activate**	**recreate**	**terminate**	**restore**	**query**	**verify**	**attach**	**detach**
`-star`	✔	✔	✔	✔	✔				
`-svp`	✔								
`-symforce`				✔					
`-v`	✔	✔	✔	✔	✔			✔	✔
`-vxfs`		✔							

a. The `-preserveTGTLocks` and `-lockid` option is not supported for `-cg` commands.

Note: For command syntax and descriptions of the `symsnap` options, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Command options with device files

With the `symsnap -file` command, you can perform snap control operations on device pairs defined in a device file. The device file includes a source device number and a target virtual device number. You must include the Symmetrix ID in the command-line argument. Table 9 lists the `symsnap` control operations and the possible options to use when targeting device pairs specified in a device file of a given Symmetrix array.

Table 9 symsnap -file control arguments and possible options

	Argument action								
Options	**create**	**activate**	**recreate**	**terminate**	**restore**	**query**	**verify**	**attach**	**detach**
`-c`, `-i`	✔	✔	✔	✔	✔	✔	✔	✔	✔
`-concurrent`			✔				✔		
`-consistent`		✔							
`-copied`							✔		
`-copyonwrite`							✔		
`-created`							✔		
`-force`	✔	✔	✔	✔	✔				
`-full`					✔				
`-multi`						✔			
`-noprompt`	✔	✔	✔	✔	✔				
`-not_ready`		✔		✔	✔				
`-ppath`		✔							
`-preaction`, `-postaction`		✔							
`-preservetgtlocks`, `-lockid`	✔	✔	✔	✔	✔				
`-rdb`, `-db`, `-dbtype`		✔							
`-restinprog`							✔		
`-restore`						✔			
`-restored`				✔			✔		
`-sid`	✔	✔		✔	✔	✔	✔	✔	✔
`-skip`	✔	✔	✔	✔					
`-star`	✔	✔	✔	✔	✔				
`-symforce`				✔					
`-v`	✔	✔	✔	✔					
`-vxfs`		✔							

Note: For command syntax and descriptions of the `symsnap` options, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

4

Performing TimeFinder/Mirror Operations

This chapter describes the business continuance model and explains how to manage and control TimeFinder/Mirror (BCV) devices using the SYMCLI in Symmetrix DMX environments running 5773 and earlier.

Note: For Symmetrix configurations running Enginuity release level 5874 and Solutions Enabler v7.0, the TimeFinder/Mirror functions described herein will be performed through TimeFinder/Clone software using a process called Clone Emulation. Clone Emulation mode makes the use of RAID-protected BCVs transparent to the TimeFinder/Mirror user.

For backward compatibility, TimeFinder/Clone Emulation mode transparently performs TimeFinder/Mirror commands and executes scripts written for Solutions Enabler up through version 6.5.2 running on Symmetrix arrays using Enginuity release levels 5773 and earlier.

Introduction to Clone Emulation mode and TimeFinder/Mirror

Clone Emulation mode is a mapping procedure that enables environments the use RAID-protected BCV devices transparent to TimeFinder/Mirror users. As previously mentioned, under Clone Emulation mode, Solutions Enabler CLI functions convert TimeFinder/Mirror commands to TimeFinder/Clone commands.

Note: For environments using Enginuity release level 5874 and later, TimeFinder/Mirror uses Clone Emulation mode for all operations. When running Enginuity 5874, any differences in operations that you need to keep in mind will be noted in their respective sections.

"TimeFinder/Clone Emulation" on page 120 contains greater detail.

The Business Continuance model

Symmetrix TimeFinder/Mirror is essentially a business continuance solution that allows the use of special *business continuance volume* (BCV) Symmetrix devices. Copies of data from a standard Symmetrix device (which are online for regular I/O operations from the host) are sent and stored on BCV devices to mirror the primary data. Uses for the BCV copies can include backup, restore, decision support, and applications testing. Each BCV device has its own host address, and is configured as a stand-alone Symmetrix device.

Business Continuance

A Business Continuance sequence first involves *associating* and *establishing* the BCV device as a mirror of a specific standard Symmetrix device. As a result, the BCV device becomes inaccessible (*Not Ready* in Figure 27) using its original device address while it is in an established pair. Once the BCV device is synchronized, you can separate (*split*) it from the standard device with which it is paired, thereby making it available again to its host for backup or other host processes through its original device address.

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

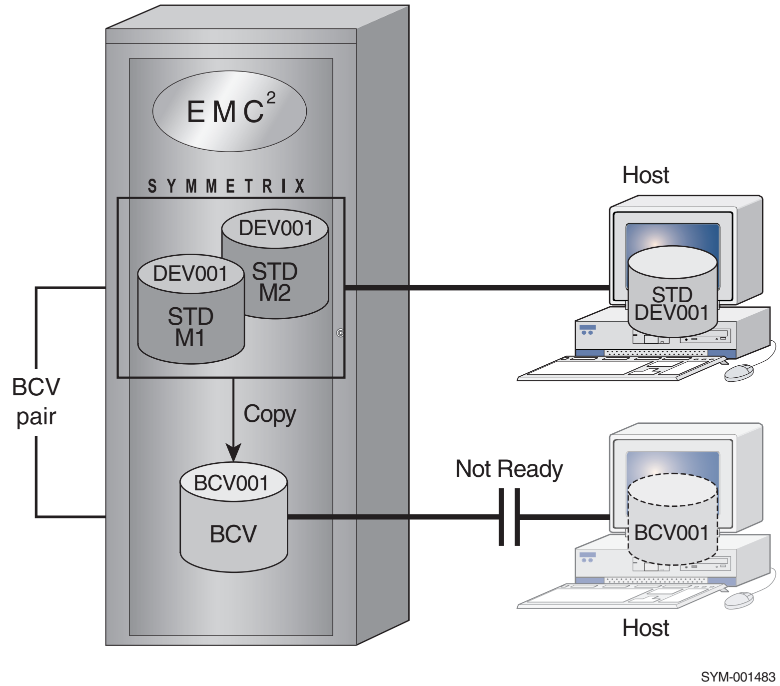

Figure 27 **Establishing a BCV pair**

A Symmetrix DMX [1] array allows up to four mirrors for each logical volume. The mirror positions are commonly designated M1, M2, M3, and M4. A single BCV can be the second, third, or fourth mirror of the standard device. In Figure 27, because standard device DEV001 is configured with two mirrors, BCV001 functions as the third mirror. A host logically views the Symmetrix M1/M2 mirrored devices as a single device.

SRDF-connected Symmetrix sites

SRDF is a Business Continuance solution that maintains a mirror image of data at the device level in Symmetrix arrays located in physically separate sites. In an SRDF configuration, the individual Symmetrix devices are designated as either a *source* or *target* to synchronize and coordinate SRDF activity.

Note: “Associating an SRDF-connected BCV device” on page 125 discusses remote operations with SRDF-connected BCVs.

Remotely mirroring the local standard

There are multiple types of SRDF-connected BCV devices. An SRDF-connected BCV can be paired with the R1 mirror or R2 mirror of the local RDF standard devices (RBCV) as shown in Figure 28.

1. Unless noted otherwise, all references Symmetrix arrays discussed in the context of TimeFinder/Mirror indicates a Symmetrix DMX running Enginuity 5773 and earlier.

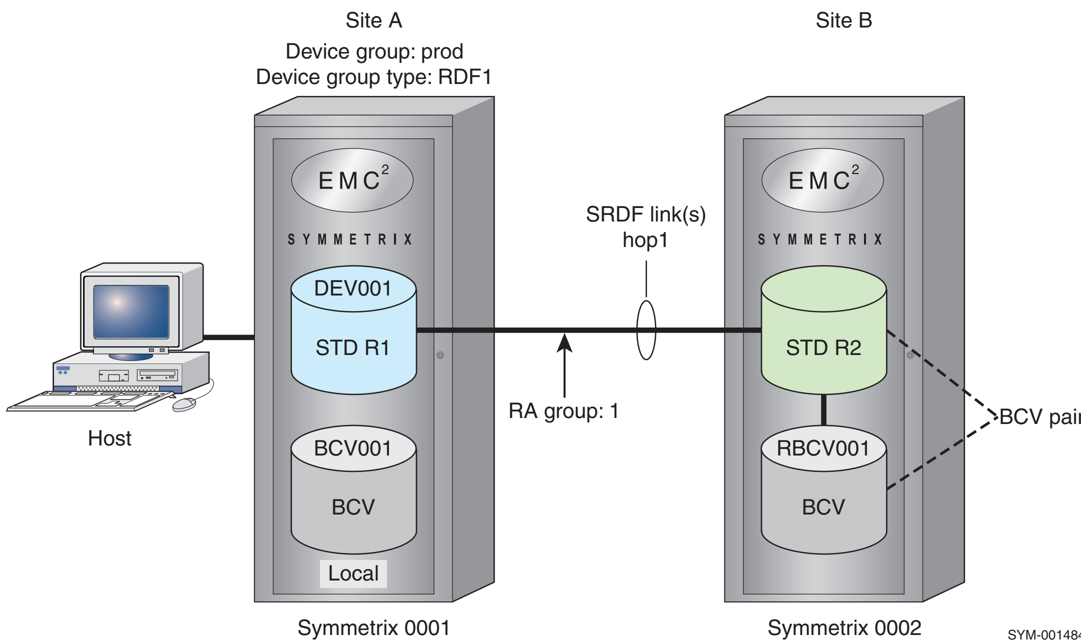

Figure 28 **SRDF: Mirroring the local standard**

Remotely mirroring the local BCVs

In another scenario, the SRDF-connected BCV can be paired with the R1 mirror or R2 mirror of the local RDF BCV device (BRBCV). In this case, the SRDF-connected BCV devices can be remotely associated with one device group of any type (Figure 29).

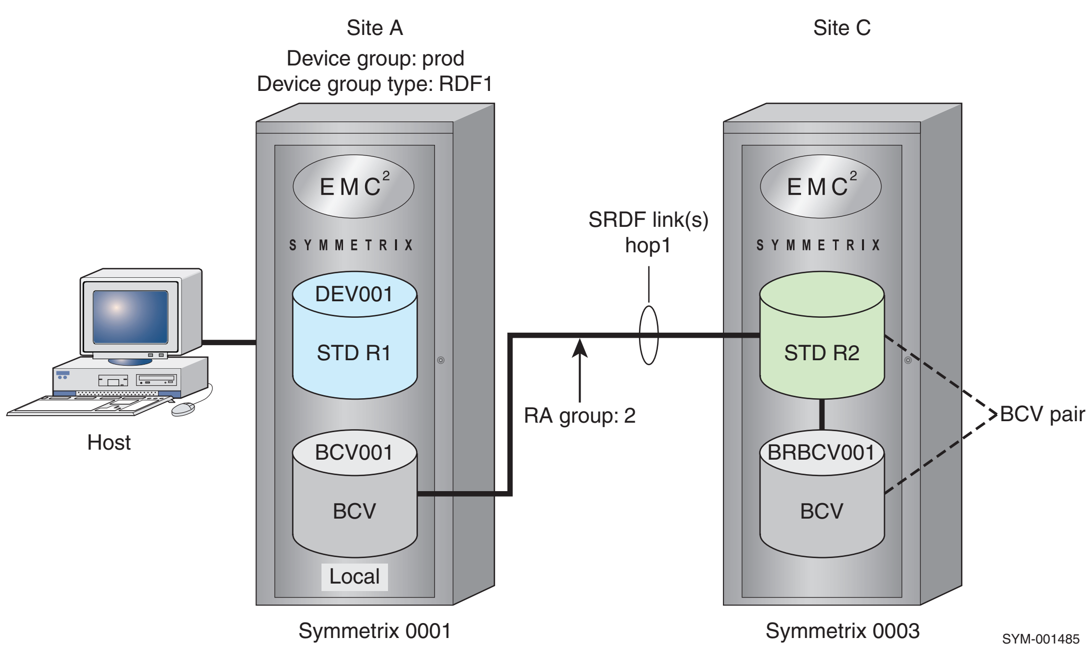

Figure 29 **SRDF: Mirroring the local BCVs**

Multihop mirroring the remote BCVs

In a multihop configuration, the SRDF-connected BCV can be paired with the R1 mirror or R2 mirror of the remote RDF BCV devices (RRBCV). In this case, the SRDF-connected BCV device can be remotely associated with one device group of type RDF1 or RDF2 (Figure 30).

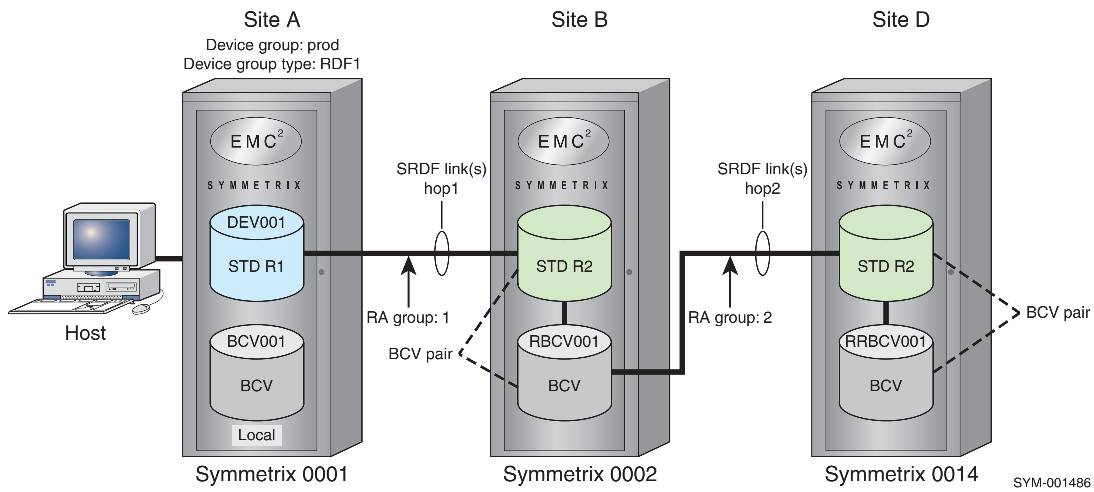

Figure 30 SRDF: Mirroring the remote BCVs

Cascaded SRDF

A Cascaded SRDF configuration consists of a primary site (site A) replicating to a secondary site (site B), which is replicating to a tertiary site (site C). This type of configuration uses a dual role R21 device on the secondary site, which acts both as an R2 device to the primary site and as an R1 device to the tertiary site.

A BCV device can be introduced into a Cascaded SRDF configuration as either of the following:

- A hop 2 BCV (2BCV) paired with the R2 partner of an R1 mirror of an R21 device (Figure 31).
- A hop 2 BCV (2BCV) paired with the R1 partner of an R2 mirror of an R21 device (Figure 32).

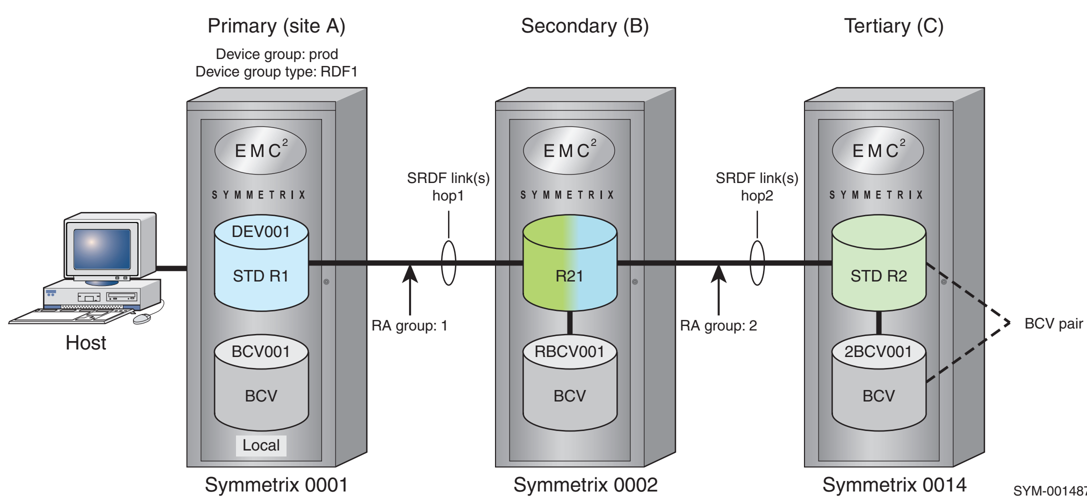

Figure 31 SRDF: Mirroring an R1 standard with Cascaded SRDF

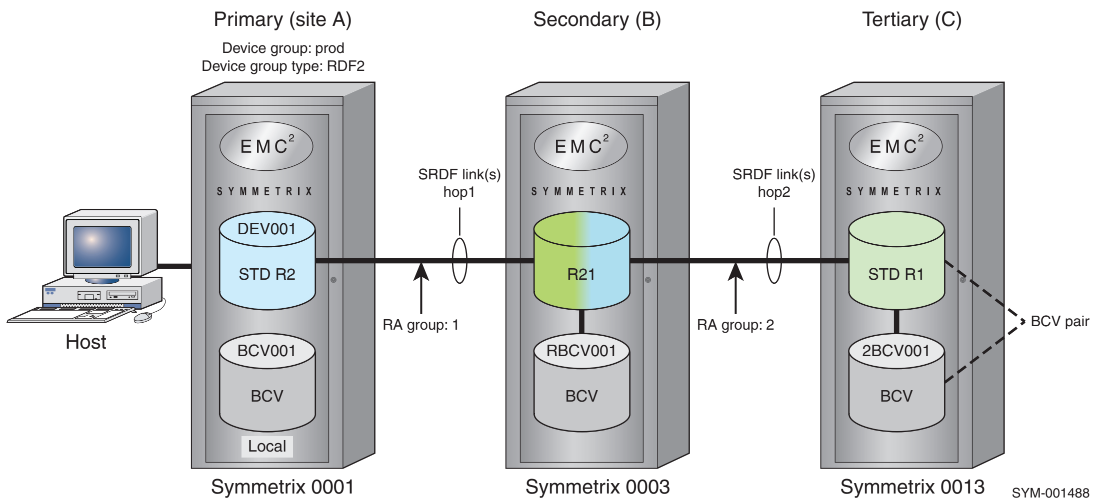

Figure 32 SRDF: Mirroring an R2 standard with Cascaded SRDF

Mirror Types

Once a BCV device is established as a mirror of a standard device, those two devices together are referred to as a ***BCV pair***. The pair consists of two types of mirrors: the standard device mirror(s) and the BCV mirror.

The standard device mirrors contain copies of the data contained in their associated standard devices. There can be up to three standard device mirrors (M1, M2, M3).

A *BCV mirror* is a standard device mirror. It can be a two-way mirror (M1, M2) that is assigned upon creation of the BCV pair.

Note: Mirroring tasks such as establish, split, and restore use the `symmir` command and are described later in this chapter.

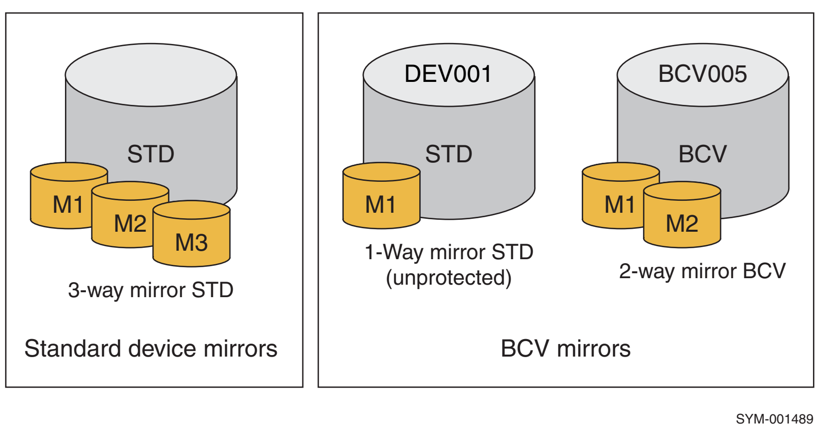

Figure 33 Mirror configuration types

TimeFinder operations overview

The TimeFinder/Mirror control operations described in this chapter require a TimeFinder or TimeFinder/Mirror license unless otherwise specified.

These management operations can safely be performed with SYMCLI on a copy of an actively changing set of the business data. Business Continuance management operations include backing up a static copy of a database, or preparing for application upgrades.

Device external locks

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

Note: The *Solutions Enabler Symmetrix Array Management CLI Product Guide* provides more information on external locks.

Skip locks

During TimeFinder/Mirror operations, you can choose to bypass the device external locks on standard devices by using the `-skip` option with the `symmir establish` and `split` commands. If the specified source devices are either all locked or all unlocked, this option will explicitly not lock those source devices.

Preserve target locks

Device external locks are used to lock BCV device pairs and devices participating in a copy session during TimeFinder BCV, Snap, and Clone operations. For target devices that have been previously locked with the same lock holder ID, you can preserve the original lock for use in performing additional TimeFinder control operations.

Use the `-preservetgtlocks` and `-lockid` options with the `symmir establish`, `restore`, and `split` commands to preserve the original device lock on target devices. You must specify the original lock holder ID number. This option causes the operation to not take out additional locks for the specified target devices.

Disallow synchronization actions

For some sites, it may be desirable to block users on a specific host from performing either an establish or restore operation on any of the Symmetrix devices. The SYMAPI_SYNC_DIRECTION parameter in the options file allows you to confine TimeFinder and SRDF operations to either just establish or restore actions. In this way, you can block a user on a host from executing a restore or an establish action using the following form:

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

where:

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

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

`BOTH` (default) does not restrict any TimeFinder or SRDF actions.

Note: The EMC Solutions Enabler Installation Guide provides more detail on the options file.

Wait for synchronization actions to complete

By default, the `symmir establish` command initiates a series of tasks that begins the synchronization of one or more BCV pairs. SYMCLI returns control to the caller while the establish operation is still in progress. The WAIT_FOR_BCV_SYNC parameter in the options file enable you to delay returning control to the caller until the establish operation (or a restore operation) is complete:

```
SYMAPI_WAIT_FOR_BCV_SYNC = TRUE | FALSE
```

where:

`TRUE` waits for the operation to complete before returning control to the caller.

`FALSE` (default) returns control to the caller before the operation completes.

Note: The *EMC Solutions Enabler Installation Guide* provides more detail on the options file.

TimeFinder/Mirror control operations

The `symbcv` command enables you to perform the following tasks on device pairs:

- Associate a BCV device with a device group or a composite group.

  Associate a BCV device or all BCV devices to the same group containing the standard devices.
- Disassociate a BCV device from a device group or a composite group.

  Disassociate the associated BCV device(s) from the specified group.
- List all BCV devices in the Symmetrix array.
- Move a BCV device from one group to another.
- Copy a BCV device from one group to another.
- Remove all BCV devices from the specified device group.

The `symmir` command enables you to perform the following mirroring tasks on standard-BCV pairs:

- Fully establish a BCV pair.

  Assign the BCV as the next available mirror of a standard Symmetrix device and copy the entire contents of the standard device to the BCV.
- Incrementally establish a BCV pair.

  Reestablish the BCV as the next available mirror of the standard device to which it was assigned before it was split.

  Synchronize the BCV device with the standard device. Any data written to the:

  - BCV device while it was split from the standard device is discarded and refreshed from the standard device.
  - Standard device while it was split is copied to the BCV device.
- Concurrently establish BCV pairs.

Establish two BCV devices (eight when using emulation mode) as concurrent mirrors of a single standard device all within the same command line. This relationship is known as a *concurrent* BCV pair. This feature allows you or an application script to instantly generate synchronized copies of the standard data.

- Split a BCV pair.

  Split the BCV from the standard Symmetrix device and make it available to the host through its separate device address.

- Fully restore from a BCV device.

  Assign the BCV device as the next available mirror of a standard device and copy the entire contents of the BCV to the standard device. The standard device does not need to be the same device originally established with the BCV.

- Incrementally restore from a BCV device.

  Reassign the BCV as the next available mirror of the standard device to which it was assigned before it was split. Synchronize the standard device with the BCV device. Any data written to the BCV while it was split from the standard device is copied to the standard device. Any updates written to the standard device while the BCV pair was split are refreshed from the BCV.

- Specify a preferred attachment of a BCV to the standard.

  This optional task marks the specified BCV device as the preferred BCV to pair with the standard device. It validates the BCV device as a valid mirror candidate and marks the BCV as the preferred device to use whenever a full establish or a full restore operation is requested. This eliminates the need to specify a device for each subsequent full establish or restore operation.

- Specify a preferred detachment of a BCV from the standard.

  This optional task unmarks the BCV device as the preferred BCV to pair with the standard device.

- Cancel the existing relationship between the specified standard and BCV device(s). This operation will put the BCV in either of the following states, depending on whether you are running TimeFinder in native or emulation mode:

TimeFinder Mode	BCV State
Native	SplitNoInc
Emulation	Never Establish

  This means that you cannot incrementally establish or incrementally restore that BCV pair again until you have performed another full establish operation or a full restore operation.

The following sections describe how to invoke these BCV control operations using the `symbcv` and the `symmir` command.

TimeFinder/Clone Emulation

Starting with Enginuity™ Version 5671, TimeFinder automatically maps a TimeFinder/Mirror command to the executable of the appropriate TimeFinder/Clone command when it encounters a BCV that is a RAID-5-or RAID-6 protected device. Under Clone Emulation mode, TimeFinder/Clone initiates the pre-copying of data.

Note: While Clone Emulation mode is primarily a RAID-5 BCV implementation, it can also be used with any other BCV protection. Clone Emulation is available with the TimeFinder/Clone license and can be used with existing TimeFinder/Mirror scripts.

When you establish a BCV pair under Clone Emulation, the `symmir establish -full` command maps to the `symclone create -precopy -differential` command. This action causes copying to begin while still checking for new writes. The `symmir split` command maps to the `symclone activate` command. This action causes the data to become available to the host as an instant point-in-time copy.

Table 10 details the mapping of TimeFinder/Mirror operations to their TimeFinder/Clone operational equivalents.

Table 10 TimeFinder commands mapped to clone operation

TimeFinder /Mirror symmir operations	TimeFinder/Clone symclone operations
FULL ESTABLISH	CREATE with pre-copy and differential
SPLIT	ACTIVATE or SPLIT
INCREMENTAL ESTABLISH	RECREATE
FULL RESTORE	FULL RESTORE
INCREMENTAL RESTORE	INCREMENTAL RESTORE
VERIFY	VERIFY
ATTACH	ATTACH
DETACH	DETACH
CANCEL	TERMINATE
PROTECT RESTORE	Default feature
PROT BCV ESTAB	Default feature[a]
QUERY	QUERY
List	List

a. Only after the completion of split is the target device fully synchronized as in a protected BCV Establish.

To operate in a mixed BCV set of RAID-5 BCVs and non-RAID-5 BCVs, you must set the Clone Emulation environment variable to ENABLED:

```
SYMCLI_CLONE_EMULATION=ENABLED
```

In a *mixed* BCV set, if Clone Emulation is *disabled* (the default), any control operation produces an error when a RAID-5 BCV is encountered.

When in Clone Emulation mode, a standard device can be paired with as many as eight concurrent BCVs (RAID-5 or any other BCV protection). Issue the `symmir` `establish -concurrent` command for the same standard device up to eight times, adding one additional BCV each time.

The following limitations apply to TimeFinder/Clone Emulation:

- The TimeFinder reverse split feature is not allowed.
- Restores will always be protected restores.
- Incremental Restore (Reverse Re-Snap) will only be accepted if all tracks were originally copied from the source prior to the restore taking place.

  Note: With Clone Emulation mode, an incremental restore will only be accepted if the devices are in a Split state and there is no active split.

- The following option file settings will be ignored:

  SYMAPI_DEFAULT_BCV_SPLIT_TYPE,
  SYMAPI_DEFAULT_BCV_RESTORE_TYPE
  SYMAPI_DEFAULT_BCV_ESTABLISH_TYPE
- Any snap or clone operation to an emulated BCV device will be rejected.
- The maximum number of BCVs that can be incrementally established with a Standard device will be eight instead of the 16 allowed by TimeFinder. SYMCLI_MAX_BCV_PAIRS can at most be eight.
- The BCV states Split-Before-Sync and Split-Before-Restore are not valid states for an emulation device. In both cases, a forced split will complete the synchronization operation.

Listing BCV devices

Configuration and status information is stored in the Symmetrix configuration database file for each device on every Symmetrix array, including BCV devices.

You can find all BCV devices on a Symmetrix array and view their physical and Symmetrix device names. In addition, you can display details about the BCV devices, including the BCV device name, the Symmetrix device name of the paired standard device, the number of invalid tracks for both the BCV device and the standard device, and the BCV pair state.

Listing host-visible BCV devices

To list all the BCV devices that are visible to the host, enter:

```
symbcv list pd
```

Listing all BCV devices

To list all the BCV devices, regardless of whether they are visible to the host, enter:

```
symbcv list dev
```

Note: The `dev` statement is optional. The truncated command `symbcv list` will return the same set of values.

Listing BCV devices with SCSI reservations

To list all the BCV devices that have SCSI reservations, regardless of whether they are visible to the host, enter:

```
symbcv -resv list
```

Listing all BCV sessions on a Symmetrix array

To list all the BCV sessions created on a Symmetrix array, use the following command:

```
symmir -sid SymmID list
```

For example, the following command lists all the BCV sessions created on Symmetrix 3264:

```
symmir list -sid 3264

Symmetrix ID: 000000003264

  Standard Device       BCV Device              State
-------------------- ----------------------- --------------
           Invalid             Invalid   GBE
Sym        Tracks    Sym       Tracks         STD <=> BCV
-------------------- ----------------------- --------------
002B                0 0E0B                0   ... Synchronized
002E                0 0E00                0   ..X Synchronized
002E                0 0E0A                0   ... Synchronized
0032                0 0E0F                0   ... Split
00FF                0 00FD                0   ... Split
0DF5                0 0DA5                0   ..X Synchronized
0DF5                0 0DA4                0   ..X Synchronized
```

```
0F70                  0 001B          3592   X.. SyncInProg
0F71                  0 001C          4496   X.. SyncInProg
0F93                  0 0DF9             0   ..X Split
1015                  0 1069             0   X.. Synchronized

Total       --------           --------
  Tracks            0              8088
  MB(s)           0.0             505.5

Legend:

(G): X = The paired BCV device is associated with a group.
     . = The paired BCV device is not associated with a group.
(B): X = The paired BCV device is splitting in the background.
     . = The paired BCV device is not splitting in the background.
(E): X = The paired BCV device is emulation mode.
     . = The paired BCV device is not emulation mode.
```

Associating BCV devices with a device group

Various compound (even multihop) remote configurations can be managed by your host using various SYMCLI control components. To perform operations on a BCV device using the SYMCLI, the BCV device must be associated with an existing device group or composite group (this is not a requirement when using a device file).

Note: For information on associating BCV devices with a composite group, refer to "Associating BCV devices with a composite group" on page 131.

When you associate a BCV device with a device group, you can assign it a logical device name. If you do not assign the BCV device a logical device name, a logical device name will be assigned automatically.

Note: Mirroring tasks such as establish, split, and restore use the `symmir` command and are described later in this chapter.

Associating a BCV device with a device group

You can associate BCV devices with a device group by using either the physical device name, or the Symmetrix device name. For example, to associate a BCV device with a physical device name of `/dev/dsk/c0t2d0s2`, to a device group named `prod`, and naming the BCV device `BCV7`, enter:

```
symbcv -g prod associate pd c0t2d0 BCV7
```

To associate a BCV device, with a Symmetrix device name of `001F`, to a device group named `prod`, naming the BCV device `BCV5`, enter:

```
symbcv -g prod associate dev 001F BCV5
```

Associating a BCV device with multiple device groups

By default, a BCV device cannot be associated with more than one device group at the same time when you are using one SYMCLI configuration database file. However, you can change this default behavior by enabling the SYMAPI_ALLOW_DEV_IN_MULT_GRPS parameter in the `options` file.

Associating all local BCV devices

You can associate all BCV devices with a device group. Only BCV devices that are not already associated with a device group will be associated.

You can either associate all BCV devices that are visible to your host within a device group (the default), or are configured in a Symmetrix array. For example, to associate all BCV devices on Symmetrix `123` with a device group `prod`, enter:

```
symbcv -g prod -sid 123 associateall dev
```

To associate all BCV devices visible to your host with the device group `prod`, enter:

```
symbcv -g prod associateall
```

In addition, you can also associate a range BCV devices that are visible to your host. For example, to associate devices `039A` through `039F` with the device group `prod`, enter:

```
symbcv -g prod associateall dev -range 039A:039F
```

Associating remote BCV devices

You can associate remote BCV devices with a device group. The following options allow all remote BCV devices of a specific type to be associated with a device group:

- `-rdf` specifies remote attached BCVs (RBCVs).
- `-bcv` specifies remote BCVs that mirror local BCVs (BRBCVs). This option must be used with the `-rdf` option.
- `-rrdf` specifies BCVs that are remotely associated with remote BCVs (RRBCVs).
- `-hop2` specifies BCV devices (2BCVs) that are remotely associated on the second hop of a Cascaded SRDF configuration.

Associating an SRDF-connected BCV device

An SRDF-connected BCV device must be associated with a device group before it can be paired with a remote standard device. Figure 34 on page 126 shows a device group of type RDF1 (there are three other device group types, RDF2, RDF21, and REGULAR, which are not shown). If the group is an RDF1 type, then the remote BCVs (RBCV) can only be paired with the R2 mirrors on the remote Symmetrix array. If the group is an RDF2 type, then the remote BCVs can only be paired with the R1 mirrors on the remote Symmetrix array. When dealing with concurrent RDF devices, you can only remotely associate with one RDF group.

For example, to associate an SRDF-connected BCV device to a device group named `prod,` and assign it a logical device name of `RBCV001`, enter:

```
symbcv -sid 123 -g prod -rdf -RDFG 1 associate dev 000B RBCV001
```

where:

`123` is the ID of the local Symmetrix array

`-rdf` specifies a remote attachment

`-RDFG 1` specifies the RDF group number (or RA group number) `1` at the local Symmetrix array through which the remote BCV is reached

`dev 000B` specifies the Symmetrix device name of the BCV on the remote Symmetrix array

In this example, the remote BCV pair is mirroring the local standard device as shown in Figure 34 on page 126.

Associating an SRDF-connected BCV mirror device

You can also associate a remote BCV pair that mirrors a local BCV device as shown in Figure 34 on page 126.

For example:

```
symbcv -sid 123 -g prod -rdf -bcv -RDFG 2 associate dev 002A BRBCV001
```

where:

`123` is the ID of the local Symmetrix array

`-rdf` specifies a remote attachment

`-bcv` specifies that it is a remote BCV that mirrors the local BCV

`-RDFG 2` specifies an RDF group number `2`, through which the remote BCV is reached

Associating devices in a compounded remote configuration

Figure 34 shows how devices in a compounded remote configuration can be associated.

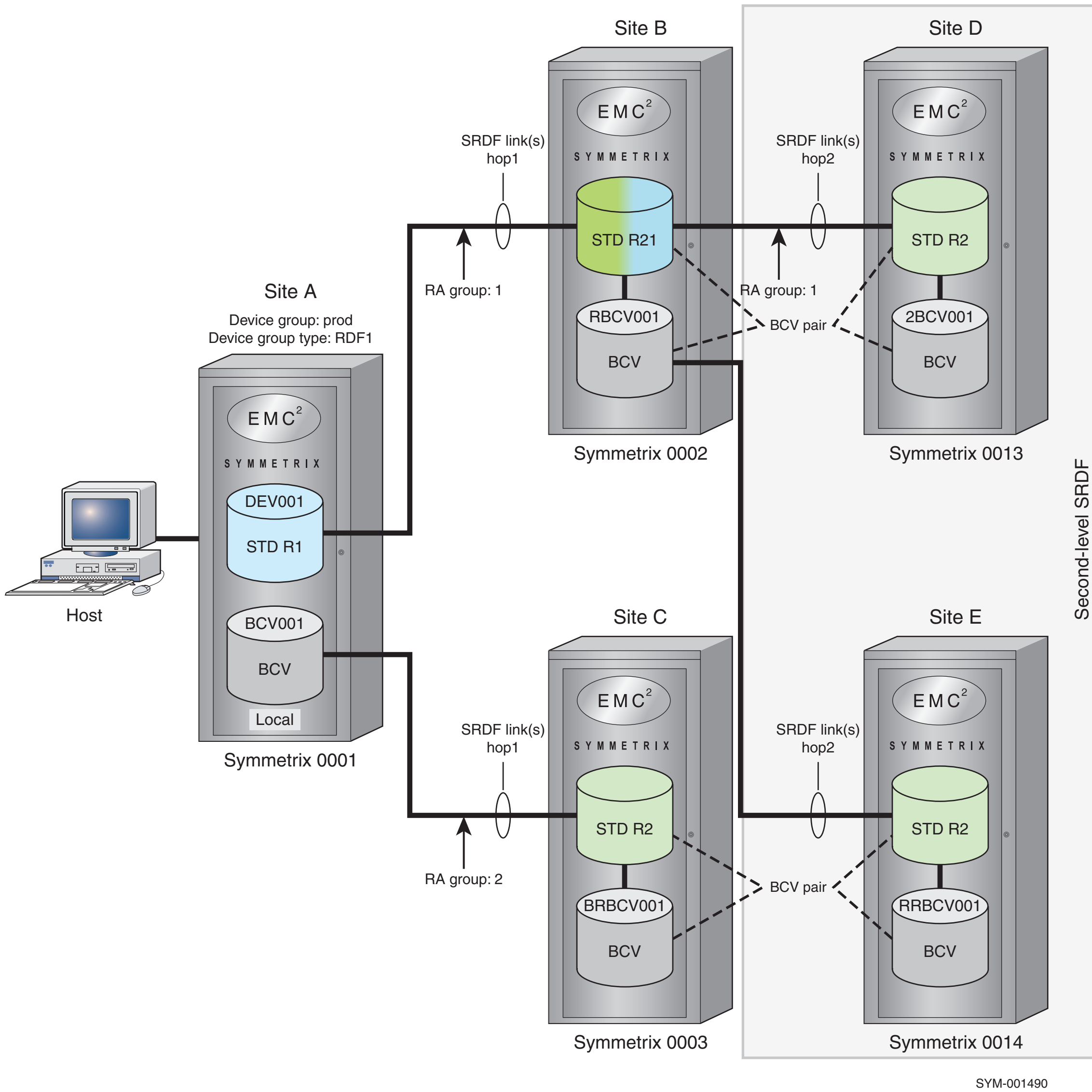

Figure 34 Compounded remote configuration example

- At site A, the SRDF links remotely mirror a local Symmetrix array.
- The remote site B functions as the remote mirror to the standard devices at site A.
- Remote site C (third site) uses an SRDF link to remotely mirror the BCV devices in the Symmetrix array at site A.
- The second-level SRDF shows how:
  - SRDF can be cascaded where remote site D (the Tertiary site) functions as a remote partner of the R21 device, which is the remote partner of the local RDF standard device.
  - SRDF can also be multihopped where remote site E functions as a remote mirror to the standard devices of site A.

Example: Creating a device group

To create the device group, add the standard device, and associate the devices shown in Figure 34 on page 126. To create the device group, `prod`, which you can then associate devices with, enter:

```
symdg create prod -type RDF1
```

This creates a device group named `prod` of type `RDF1`. Next, the standard, device `001` of Symmetrix `0001` is added to group `prod` and given a logical name of `DEV001`, by entering:

```
symld -g prod add dev 0001 -sid 0001 DEV001
```

Once the STD device has been added to the group, the BCV device `000A` is added to group `prod` and given a logical name of `BCV001`, by entering:

```
symbcv -g prod associate dev 000A BCV001
```

Once the BCV device has been added to the group, the remote BCV (RBCV) device `000B` is added to group `prod` and given a logical name of `RBCV001`, by entering:

```
symbcv -g prod associate dev 000B -rdf RBCV001
```

The remote BCV of the local BCV (BRBCV) device `00C` is added to group `prod` and given a logical name of `BRBCV001`, by entering:

```
symbcv -g prod associate dev 000C -bcv -rdf BRBCV001
```

The remote BCV (2BCV) on the second hop of a Cascaded SRDF configuration device `000D` is added to group `prod` and given a logical name of `2BCV001`, by entering:

```
symbcv -g prod associate dev 000D -hop2 2BCV001
```

The remote BCV of the remote BCV (RRBCV) device `000E` is added to group `prod` and given a logical name of `RRBCV001`, by entering:

```
symbcv -g prod associate dev 000E -rrdf RRBCV001
```

At this point, all these devices can be controlled with the TimeFinder and SRDF commands.

Note: In the previous example, the devices are associated to a device group beginning with the nearest Hop (DEV001) and proceeding to the furthest HOP device (in this case RRBCV001). If you choose to associate a device to a group in a different order (such as RRBCV001 prior to DEV001), then you must provide additional configuration options to define the RDF link relationship to the local device, such as: RA group numbers (`-RDFG`), remote RA group numbers (`-remote_rdfg`), and Symmetrix ID (`-sid`).

Disassociating BCV devices from a device group

Once a BCV device has been associated with a device group, you can disassociate it when the BCV device is in a state that allows it to be disassociated.

You can disassociate a BCV device by using either the physical device name, the Symmetrix device name, or the logical device name. For example, to disassociate a BCV device named `BCV7` from the `prod` device group, enter:

```
symbcv -g prod disassociate ld BCV7
```

To disassociate a BCV device on Symmetrix device `000F` from the `prod` device group, enter:

```
symbcv -g prod disassociate dev 000F
```

If the BCV device that you want to disassociate from a device group is currently paired via TimeFinder, TimeFinder/Snap, or TimeFinder/Clone, the `-force` option must be used.

Note: When you use the `-force` option, SYMCLI does not access the Symmetrix array (an offline operation). It disassociates the device from the device group without access to the Symmetrix array.

Disassociating an SRDF-connected BCV pair

An SRDF-connected BCV device can be disassociated from a device group.

You can disassociate a BCV device using either the Symmetrix device name or the logical device name. For example, to disassociate a remote BCV device that has a logical device name of `RBCV001` from a device group named `SAMPLE1`, enter:

```
symbcv -g SAMPLE1 -rdf disassociate ld RBCV001
```

To disassociate a remote BCV device that has a Symmetrix device name of `002B` from a device group named `SAMPLE1`, enter:

```
symbcv -g SAMPLE1 -rdf disassociate dev 002B
```

If you are attempting to disassociate a remote BCV device from a device group, and the BCV device is in the synchronized, restored, or transient BCV state, you must use the `-force` option.

Note: When you use the `-force` option, SYMCLI does not access the Symmetrix array (an offline operation). It disassociates the device from the device group without access to the Symmetrix array.

Disassociating a remote SRDF-connected BCV pair

You can disassociate a remote BCV pair that mirrors the local BCV device, which is shown associated in Figure 34 on page 126.

For example:

```
symbcv -sid 123 -g SAMPLE1 -rdf -bcv disassociate dev 002A
```

where:

`123` is the ID of the local Symmetrix array.

`-rdf` specifies a remote attachment.

`-bcv` specifies that it is a remote BCV that mirrors the local BCV.

Disassociating a remote BCV of a remote BCV pair

You can disassociate a remote BCV that mirrors the remote BCV device, which is shown associated in Figure 34 on page 126. This option can be used to disassociate a BCV device that is accessible via SRDF links two hops away. The group must be an RDF group.

For example:

```
symbcv -sid 123 -g SAMPLE1 -rrdf disassociate dev 002A
```

where:

`123` is the ID of the local Symmetrix array.

`-rrdf` specifies the BCV is being remotely disassociated with a remote BCV in the group.

Disassociating a remote BCV on the second hop of a Cascaded SRDF configuration

You can disassociate a remote BCV on the second hop of a Cascaded SRDF configuration, which is shown associated in Figure 34 on page 126. This option can be used to disassociate a BCV device that is accessible via SRDF links two hops away. The group must be an RDF group.

For example:

```
symbcv -sid 123 -g SAMPLE1 -hop2 disassociate dev 002A
```

where:

`123` is the ID of the local Symmetrix array

`-hop2` specifies the BCV is in the second hop of a Cascaded SRDF.

Moving BCV devices from one device group to another device group

The `symbcv` command can be used to move one or all BCV devices from one existing device group to another existing device group. The source and destination groups can be of different types. When moving a BCV device from one group to another, you can choose to have the device's logical name renamed to the default naming convention of the destination group. This helps to avoid naming conflicts that may be encountered. Various other options are available to limit a move to the devices that meet a specified set of criteria, which can be specified along with the `moveall` action. For a full description of these options, refer to *symbcv* in the *Solutions Enabler Symmetrix CLI Command Reference*.

Moving a specific device

To move BCV device `BCV001` from device group `prod` to the destination group `NewGroup` and rename the moved device, enter:

```
symbcv -g prod move ld BCV001 NewGroup -rename
```

Moving all BCV devices

You can move all local BCV devices from one device group to another device group. The source and destination groups can be of different types.

To move all BCV devices that are visible to your host from device group `prod` to device group `NewGroup`, enter:

```
symbcv -g prod moveall NewGroup
```

Moving remote BCV devices

Remote BCV devices can be moved from one device group to another device group. The following options allow all remote BCV devices of a specific type to be moved:

- `-rdf` specifies remote attached BCVs (RBCVs).
- `-bcv` specifies remote BCVs that mirror local BCVs (BRBCVs). This option must be used with the `-rdf` option.
- `-rrdf` specifies BCVs that are remotely associated with remote BCVs (RRBCVs).
- `-hop2` specifies BCV devices (2BCVs) that are remotely associated on the second hop of a Cascaded SRDF configuration.

Removing devices

The `symbcv` command also contains a remove all action (`rmall`), which will remove all devices meeting the specified criteria from the specified device group.

Managing BCV devices with composite groups

When adding BCV devices to a composite group, consider the following:

- The user must specify a local Symmetrix array ID.
- If no devices have been added to the composite group yet, and if there is only one RDFG on the Symmetrix array, then that RDFG is assumed; otherwise, you must specify a RDFG when adding remote BCVs.
- The `-host`, `-SA`, and `-P` parameters are only valid when associating local BCV devices.
- If `-bcv` is specified, the BCV list affected is the BRBCV list (or the BCV of the remote BCV device).

  Note: `-rdf` is required when `-bcv` is specified.

- If `-rdf` is specified and `-bcv` is not specified, the BCV list affected is the RBCV list.
- If `-rrdf` is specified, the BCV list affected is the RRBCV list.
- If `-hop2` and `-remote_rdfg` are specified, the BCV list affected is the 2BCV list.
- If `-rdf` and `-bcv` and `-rrdf` are not specified, the BCV list affected is the BCV list.

Associating BCV devices with a composite group

Use the following syntax to associate a BCV using the given Symmetrix device name to the composite group:

```
symbcv -cg CgName -sid SymmID [[-rdf [-bcv]] | [-rrdf] | [-hop2]]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   associate dev SymDevName [LdevName]
```

Use the following syntax to associate all BCV devices for the *SymmID* to the composite group:

```
symbcv -cg CgName -sid SymmID [[-rdf [-bcv]] | [-rrdf] | [-hop2]]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   associateall [dev|-host HostName] [-SA <#|ALL>] [-P #] [-CAP #]
   [-N #] [-R1|-NOR1] [-R2|-NOR2]
   [-RANGE <SymDevStart>:<SymDevEnd>]
```

Disassociating BCV devices from a composite group

Use the following syntax to disassociate a BCV using the Symmetrix device name:

```
symbcv -cg CgName -sid SymmID [[-rdf [-bcv]] | [-rrdf] | [-hop2]]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   disassociate dev SymDevName [-force]
```

Moving BCV devices to a composite group

Use the following syntax to move the BCV that has the given Symmetrix device name to the destination composite group:

```
symbcv -cg CgName -sid SymmID [[-rdf [-bcv]] | [-rrdf] | [-hop2]]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   move dev SymDevName DestCgName [-force]
```

Fully establishing BCV pairs

After configuration and initialization of a Symmetrix array, BCV devices contain no data. The BCV devices, like the standard devices, have unique host addresses and are online and ready to the hosts to which they are connected. The full establish must be used the first time the standard devices are paired with BCV devices.

Figure 35 illustrates the initial Symmetrix configuration prior to performing any TimeFinder BCV operations. In this figure, the host views the Symmetrix M1/M2 mirrored pair as a single standard device (DEV001) and the BCV device as BCV001.

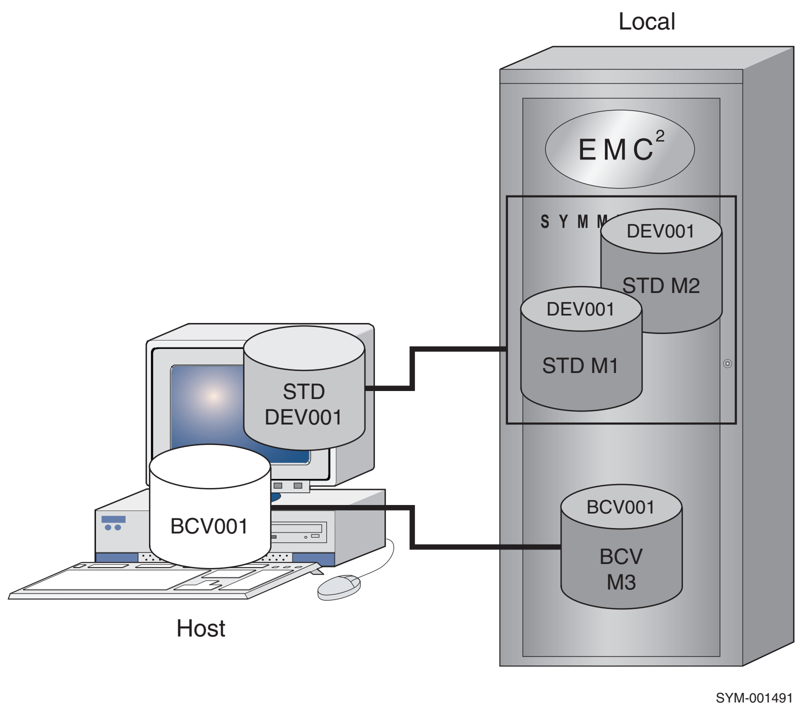

Figure 35 Initial BCV configuration

To obtain a copy of the data on a standard Symmetrix device, a BCV pair must be established. A BCV pair consists of a BCV device and a standard device. The standard device can have various mirror structures (unprotected, two-way or three-way mirrored, RAID, RAID with SRDF), as long as the number of mirrors does not exceed three. This constraint is in place because establishing a BCV pair requires assigning the BCV device as the next available mirror of the standard device.

Since there is a maximum of four mirrors allowed per device in the Symmetrix array, a device already having four mirrors is not able to accommodate another one.

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

```
symmir -g DgName -full establish
symmir -cg CgName -full establish
symmir -f[ile] FileName -full establish
```

Note: For information on creating a device file, refer to "Device file" on page 216.

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

```
symmir -g prod -full establish
```

To initiate a full establish on one BCV pair, `DEV001`, in the `prod` device group, enter:

```
symmir -g prod -full establish DEV001
```

To initiate a full establish on more than one BCV pair (list) in the `prod` device group with one command, enter:

```
symmir -g prod -full establish
          DEV001 BCV ld BCV005
          DEV002 BCV ld BCV007
          DEV003 BCV ld BCV008
```

Figure 36 illustrates the full establish of a BCV pair.

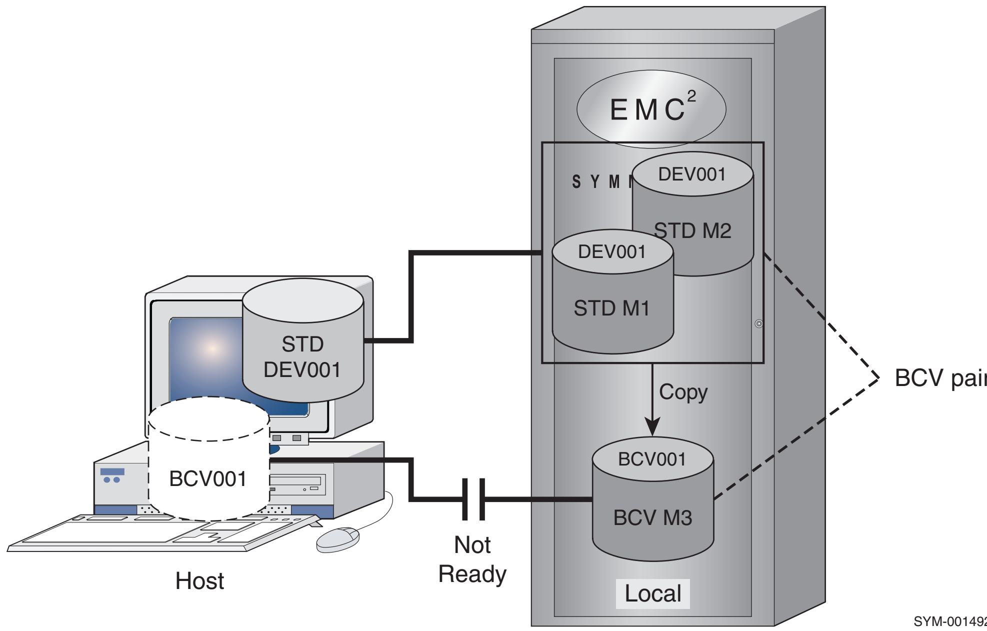

Figure 36 Fully establishing a BCV pair

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

- Command validity is checked. For example, the Symmetrix array makes sure that both the standard device and the BCV device are the same size, the device specified as the BCV has the BCV attribute, the standard device does not already have a BCV device assigned to it, and so on.

  If the standard device is a meta head device, then the BCV must also share the same meta device properties. All meta members will be implicitly established along with the meta head device.

- The BCV device is set as Not Ready to the host.
- The BCV device is assigned as the next available mirror of the standard device. A BCV can be the second, third, or fourth mirror of the standard device. For example, in Figure 36, it is the third mirror (M3).
- The contents of the standard device are copied to the BCV. For example, in Figure 36, the BCV device receives its data from both the first fully valid mirror of the source.

The BCV pair is synchronized when the standard device mirrors and the BCV mirror contain identical data.

Note: The BCV device is not available for host use during the time that it is assigned as a BCV mirror on a standard device. However, any new data written to the standard device is copied to the BCV device while the BCV pair exists.

Specifying the default method for establishing BCV Pairs

When specifying the default method for establishing BCV pairs, you can either set it at the system level by using an options parameter, or at the user level by using an environment variable.

Note: User level settings (environment variables) will override system level settings (options file parameters).

CAUTION

Because of the load that establish operations place on a Symmetrix array, you should only change these settings under the direction of EMC. Please contact your EMC representative before changing these settings.

Specifying at the system level

To specify the default method for establishing BCV pairs at the system level, set the SYMAPI_DEFAULT_BCV_ESTABLISH_TYPE parameter in the `options` file. Possible values are:

- SINGULAR specifies to issue an establish to a single device, including a meta member, at a time. This method allows other tasks access to the Symmetrix array when doing a large number of establishes.
- PARALLEL (default) specifies to issue an establish to each servicing disk adapter (DA) in parallel, and then wait for a DA to finish before issuing another establish to that DA.
- SERIAL specifies to issue establishes as fast as the Global Special Task (GST) queue can handle them. However, all members of a meta must be established before continuing to the next meta or device. This is the default method when using meta devices.

Note: The *EMC Solutions Enabler Installation Guide* contains information on changing the `options` file parameters.

Specifying at the user level

To specify the default method for establishing BCV pairs at the user level, set the SYMCLI_BCV_EST_TYPE environment variable. Possible values for this variable are SINGULAR, PARALLEL, or SERIAL, as described earlier on this page.

When specifying the default method as SINGULAR or PARALLEL, you can also set the SYMCLI_BCV_EST_DELAY environment variable to specify how long to wait between sending commands to the Symmetrix array. Possible value for this variable range from 0 to 30, with 0 being the default setting.

Note: The *EMC Solutions Enabler CLI Command Reference* contains information on changing environment variables.

Instantly establishing multiple BCV pairs

The Multi/Instant Establish option improves the performance of a typical establish operation by submitting multiple BCV pairs in a single system call to be established instantly. This feature requires Enginuity Version 5671 or greater and will be ignored on previous Enginuity versions.

You can enable (default)/disable this feature with the SYMAPI_TF_MULTI_ESTAB_REST environment variable. Setting the SYMAPI_DEFAULT_BCV_ESTABLISH_TYPE parameter to SERIAL or SINGULAR will cause this option to be ignored.

Establishing multiple BCVs with a single standard device

You can fully establish (at different times) up to 16 BCV devices (8 when using emulation mode) associated with a single standard device. The BCV devices must also be associated to the same device group.

Note: Using the environment variable SYMCLI_MAX_BCV_PAIRS, the maximum number of pairs (established or restored) can be adjusted between 1 to 16 BCV devices.

With this feature, standard devices retain a relationship with multiple BCVs as long as those BCVs do not become paired with another standard. Here, the information about changed tracks is saved for a split BCV device when another BCV device is subsequently established and split from the same standard device. By invoking a series of `split/-full establish` commands over time (as shown in Figure 37), a multi-BCV environment becomes established that retains progressive historical images of the specified standard.

Note: When the maximum number of multi-BCV pairs is reached, you can alter the BCV pair *cancel policy* that controls the round-robin device usage as you fully establish the next device beyond the set maximum pair count. Using environment variable SYMCLI_BCV_PAIR_POLICY, you can cancel the incremental relationship between the STD and the oldest BCV (default), cancel the newest, or don't cancel any BCV in the set.

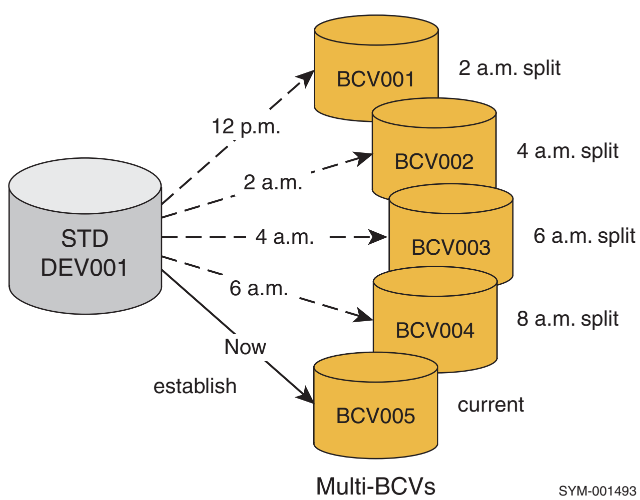

Figure 37 **Establishing a multi-BCV environment**

For example, to fully establish the standard with BCV005, enter:

```
symmir -g prod split DEV001
symmir -g prod -full establish DEV001 BCV ld BCV005
```

In this environment, you can specify any one of these older BCVs to incrementally restore the standard back to a historical copy (refer to "Incrementally restoring multiple BCV pairs" on page 182).

Canceling a multi-BCV relationship

You can completely cancel the incremental relationship between the STD and any one of the split multi-BCV devices from the set using the `cancel` operation. This operation will put the BCV in either of the following states, depending on whether you are running TimeFinder in native or emulation mode:

TimeFinder Mode	BCV State
Native	SplitNoInc
Emulation	Never Establish

You cannot incrementally establish or incrementally restore that BCV pair again until you have performed another full establish operation or a full restore operation.

For example, to remove BCV001 from the set, enter:

```
symmir -g prod cancel DEV001 BCV ld BCV001
```

As shown in Figure 38, once you cancel the specified BCV, all records of track changes between the STD and the canceled BCV are destroyed.

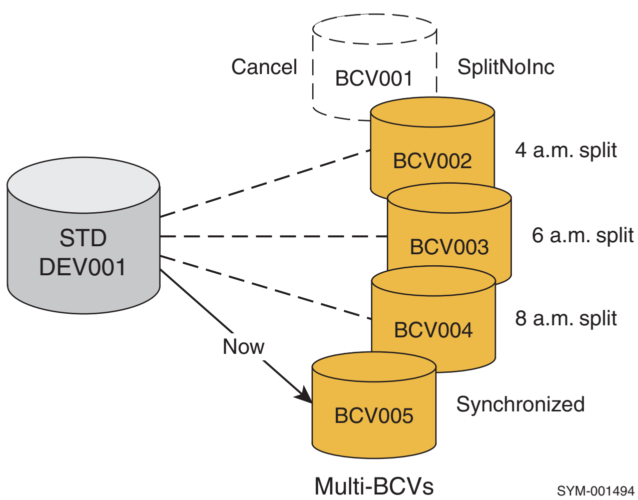

Figure 38 Canceling a multi-BCV

Establishing SRDF-connected BCV pairs

You can specify an `establish` action to a remote Symmetrix using the RDF flag (`-rdf`), which fully establishes the remote BCV pairs.

For example, Figure 39 illustrates a full establish operation in the remote Symmetrix array at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf -full establish
```

To initiate a full establish on one remote BCV pair, `DEV001`, in the `prod` group, enter:

```
symmir -g prod -rdf -full establish DEV001 bcv ld RBCV001
```

In this case, the flag indicates that the BCV device being established is an SRDF-connected BCV device, which will be established with the remote mirror of the local RDF standard device.

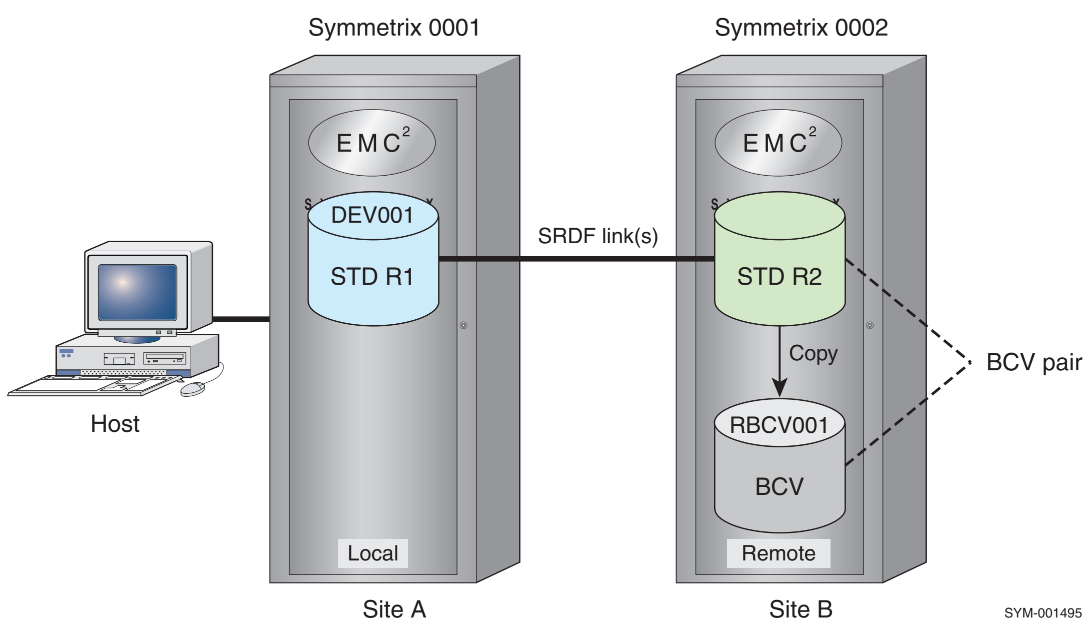

Figure 39 Establishing the remote BCV pair (mirroring the local STD)

Figure 40 illustrates a full establish operation in the remote Symmetrix array at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv -full establish
```

To initiate a full establish on one remote BCV pair, `BCV001`, in the `prod` group, enter:

```
symmir -g prod -rdf -bcv -full establish BCV001 BCV ld BRBCV001
```

In this case, the `-rdf` parameter indicates that the BCV device being established is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local R1 BCV device.

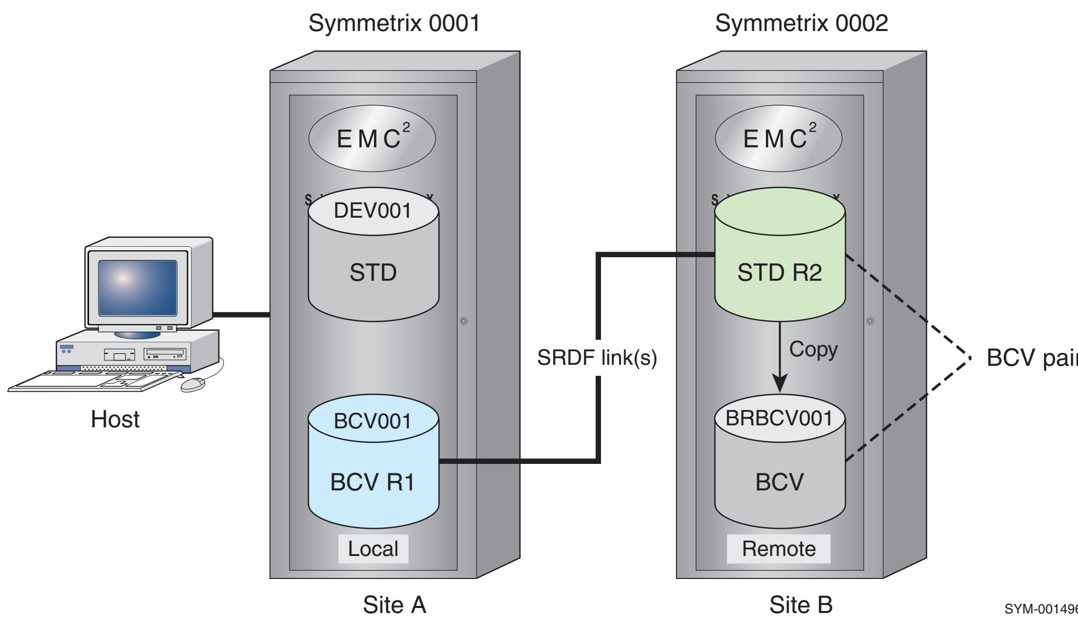

Figure 40 Establishing the remote BCV pair (mirroring the local BCV)

Establishing second-level remote BCV pairs

You can specify an `establish` action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which fully establishes second-level remote BCV pairs. Figure 41 illustrates a full establish operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv -full establish
```

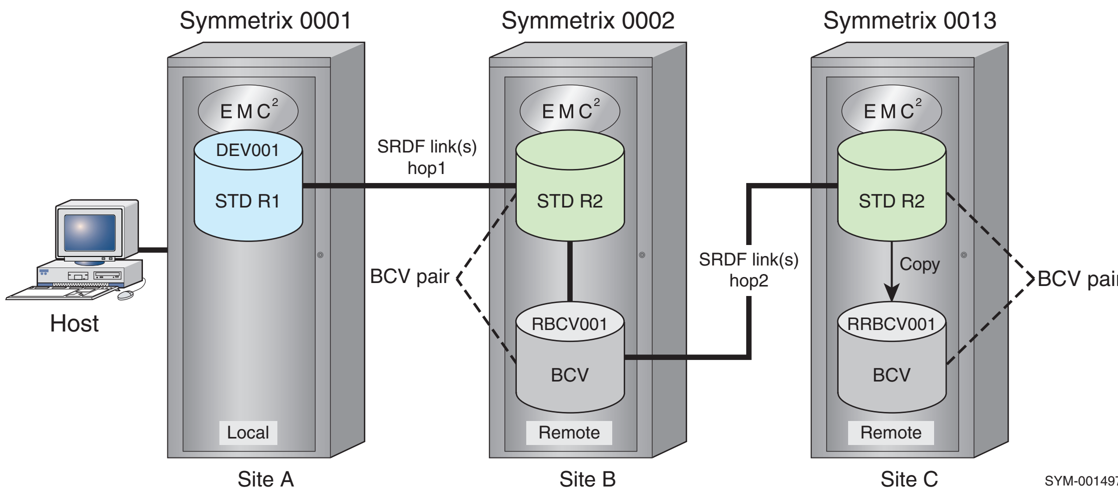

Figure 41 Establishing the remote hop 2 BCV pair (mirroring the local BCV)

To initiate a full establish on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv -full establish RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being established is a second HOP SRDF-connected BCV device, which will be established with the remote standard mirror of the remote BCV device.

Establishing hop 2 BCV pairs in a Cascaded SRDF configuration

You can specify an `establish` action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 42 illustrates a full establish operation in the remote Symmetrix at the tertiary site (C) when the hop 2 flag is specified with the following command:

```
symmir -g prod -hop2 establish -full
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being established with the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

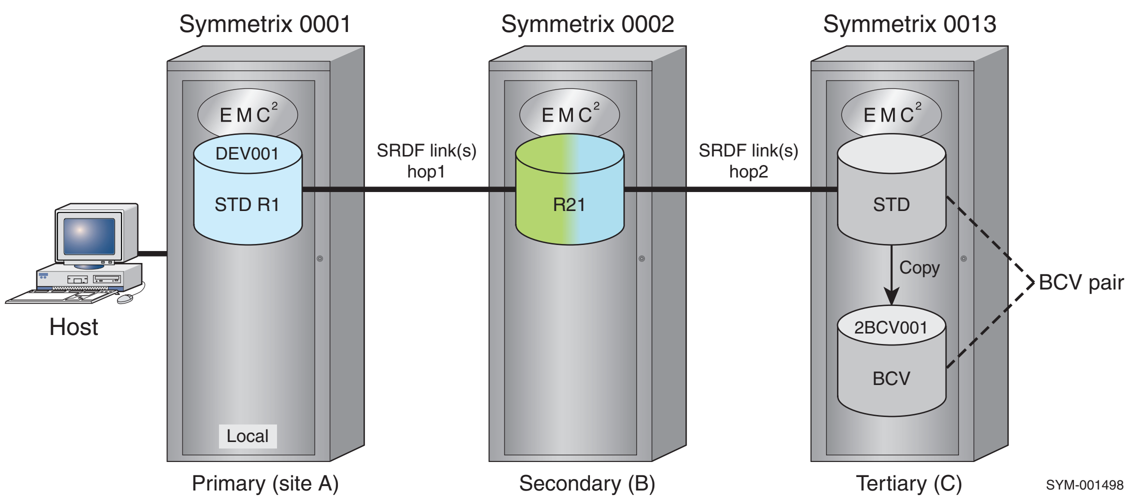

Figure 42 Establishing the hop 2 BCV pair in a Cascaded SRDF configuration

Using cascaded Clone Emulation to cascaded clone

In environments running Enginuity release level 5874 with Solutions Enabler V7.0, the target device of a Clone Emulation session can be used as the source for one or more clone sessions and vice versa:

- Clone session to a Clone Emulation session
- Clone Emulation session to a clone session

Clone session to a Clone Emulation session

In this configuration, the source device A to target device B is a TimeFinder/Clone session, and source device B to target device C (BCV) is a TimeFinder/Clone Emulation session.

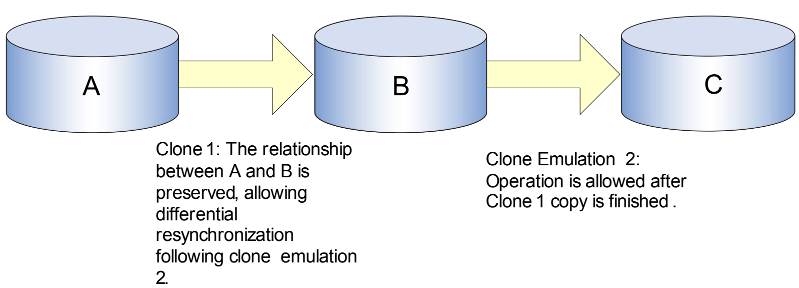

Figure 43 Cascaded Clone to a cascaded Clone Emulation session

Table 11 lists the Clone to Clone Emulation target session states.

Note: When A→B session is SyncInProgress or RestoreInProgress, the –symforce flag is required when performing a B→C split operation.

Table 11 Clone and B to C TimeFinder/Clone Emulation states

B→C session state	Clone A → Clone B → Clone C session state						
	A→B No session	A→B Created, Recreated	A→B Precopy	A→B CopyInProg CopyOnAccess CopyOnWrite	A→B Copied Split	A→B RestoreInProg	A→B Restored
B→C No session	Create A→B Full Establish A→B Full Restore A→B Full Establish B→C	Activate A→B Set Mode A→B	Activate A→B Set Mode A→B	Recreate A→B Establish A→B Set Mode A→B	Recreate A→B Establish A→B Restore A→B Set Mode A→B	Full Establish B→C	Split A→B Full Establish B→C
B→C Synchronized SyncInProgress	Create A→B Full Establish A→B Full Restore A→B Split B→C	Activate A→B Set Mode A→B	Activate A→B Set Mode A→B	Recreate A→B Establish A→B Set Mode A→B	Recreate A→B Establish A→B Restore A→B Set Mode A→B Split B→C	Split B→C	Split A→B Split B→C
B→C Split (BG Split In Progress)	Full Restore A→B Establish B→C	Activate A→B Set Mode A→B (no precopy)	Not proper state	Set Mode A→B (no precopy)	Restore A→B Set Mode A→B (no precopy)	Establish B→C	Split A→B Establish B→C
B→C Split (BG split done)	Create A→B Full Establish A→B Full Restore A→B Establish B→C Restore B→C	Activate A→B Set Mode A→B	Activate A→B Set Mode A→B	Recreate A→B Establish A→B Set Mode A→B	Recreate A→B Establish A→B Restore A→B Set Mode A→B	Establish B→C	Split A→B Establish B→C
B→C RestoreInProg		Not proper state	Not proper state	Not proper state	Not proper state	Not proper state	Not proper state
B→C Restored	Full Restore A→B Split B→C	Not proper state	Not proper state	Not proper state	Not proper state	Split B→C	Split B→C

Clone Emulation session to a clone target session

In this configuration, the source device A to target device B (BCV) is a TimeFinder/Clone Emulation session, and source device B to target device C is a TimeFinder/Clone session. Table 12 lists the Clone Emulation to Clone target session states.

Note: When A→B session SyncInProgress or RestoreInProgress, –symforce flag is required on A→B split.

Table 12 Clone Emulation and clone target session states (page 1 of 2)

B→C session state	Clone A → Clone B → Clone C session state					
	A→B No session	A→B Synchronized SyncInProgress	A→B Split (BG Split In Progress)	A→B Split (BG split done)	A→B RestoreInProg	A→B Restored
B→C No session	Full Establish A→B Create B→C Full Establish B→C Full Restore B→C	Split A→B	Establish A→B	Establish A→B Restore A→B Create B→C Full Establish B→C		Split A→B

Table 12 Clone Emulation and clone target session states (page 2 of 2)

B->C Created Recreated	Activate B->C Set Mode B->C	Not proper state	Not proper state	Activate B->C Set Mode B->C	Not proper state	Not proper state
B->C Precopy	Activate B->C Set Mode B->C	Not proper state	Not proper state	Activate B->C Set Mode B->C	Not proper state	Not proper state
B->C CopyInProg CopyOnAccess CopyOnWrite	Recreate B->C Establish B->C Set Mode B->C	Not proper state	Not proper state	Recreate B->C Establish B->C Set Mode B->C	Not proper state	Not proper state
B->C Copied Split	Recreate B->C Establish B->C Restore B->C Set Mode B->C	Not proper state	Not proper state	Recreate B->C Establish B->C Set Mode B->C	Not proper state	Not proper state
B->C RestoreInProg		Not proper state	Not proper state	Not proper state	Not proper state	Not proper state
B->C Restored	Split B->C	Not proper state	Not proper state	Not proper state	Not proper state	Not proper state

Establishing concurrent BCV pairs

When you establish a BCV device as a mirror of a standard device, that relationship is known as a BCV pair. When you sequentially establish/split/establish a number of BCV devices over time with a specified standard, that is known as a multi-BCV relationship.

You can establish two BCV devices (eight when using emulation mode) as concurrent mirrors of a single standard device all within the same `symmir` command line. This relationship is known as a *concurrent* BCV pair. This feature allows you or an application script to instantly generate two synchronized copies of the standard data. When the two BCVs are split from the standard, the BCV's hosts can access the data on either BCV.

When establishing concurrent BCV pairs, you can either specify the BCVs, or use the `-concurrent` option to allow the Symmetrix array to select suitable (in terms of size, emulation, etc.) BCV(s) from the available BCV list.

Example: Specifying the BCVs

To concurrently establish `BCV001` and `BCV002` with standard `DEV012` in device group `CncGrp`, enter:

```
symmir -g CncGrp establish -full DEV012 bcv ld BCV001
                                 DEV012 bcv ld BCV002
```

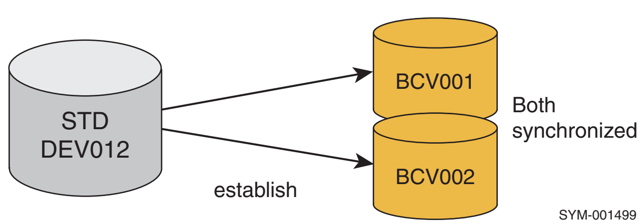

Figure 44 **Establish concurrent BCV pairs**

After the concurrent BCVs become split at some point, you can then concurrently resynchronize the BCVs with an incremental establish:

```
symmir -g CncGrp establish DEV012 bcv ld BCV001
                           DEV012 bcv ld BCV002
```

While these examples pair both BCVs at the same time, you can also establish the first BCV device now and the second one later. In either case, the concurrent BCVs become synchronized to the standard and remain that way until you split the BCVs from the standard.

The following is a valid concurrent BCV pair example, provided there is no split action between these commands:

```
symmir -g CncGrp establish -full DEV012 bcv ld BCV001
            .
            .
            .
symmir -g CncGrp establish -full DEV012 bcv ld BCV002
```

Example: Using the -concurrent Option

To instruct the Symmetrix array to select suitable BCV(s) to concurrently establish with standard `DEV012` in device group `CncGrp`, enter:

```
symmir -g CncGrp establish -full DEV012 -concurrent
```

In this case, if the standard is already synchronized with one BCV, the Symmetrix array will synchronize one other BCV with it. If the standard device is not yet synchronized with a BCV, the Symmetrix array will still only synchronize one BCV with it.

Performing a protected BCV establish (moving all mirrors)

The protected BCV establish (`-protbcvest`) option (also called the moveall establish) can be used with the establish action to move all mirrors of locally mirrored BCV devices to join the mirrors of the standard device. This protected BCV moving mirror feature (starting with Enginuity Version 5670) improves the availability of the BCV mirrors, particularly after a split operation where all BCV mirrors are instantly synchronized. This feature, also known as instant BCV mirror sync, is available with full or incremental establish or restore operations. This feature is only useful in a native TimeFinder environment.

Note: This feature is not supported if the standard device is a dynamic concurrent SRDF device or if TimeFinder is in emulation mode.

Figure 45 compares the Symmetrix array behavior between a normal and a moving protected establish action.

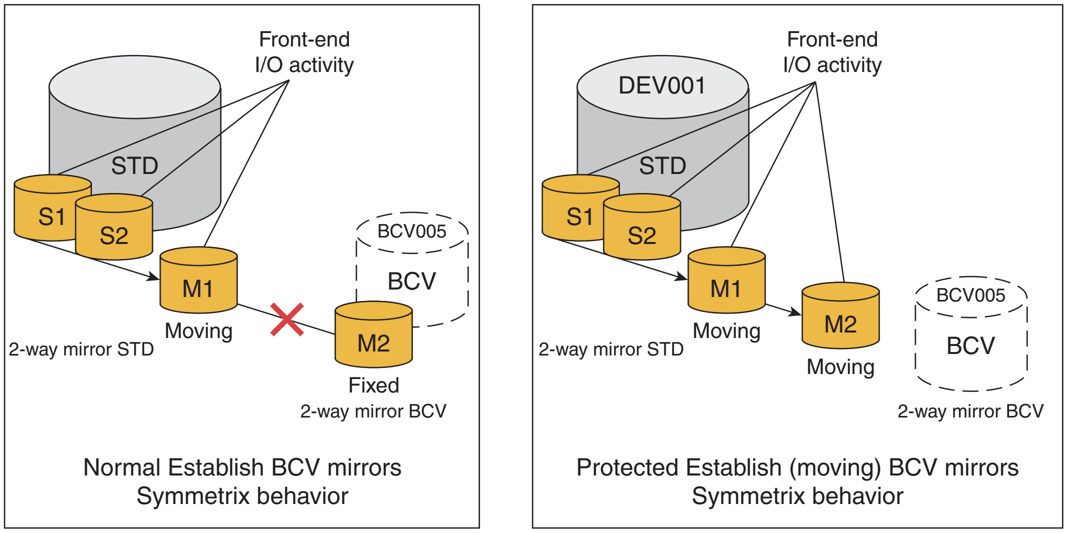

Figure 45 Establishing two-way BCV mirrors with protected establish

In a 2-way BCV mirror configuration for a normal establish, M2 is fixed and can only be updated from M1 after a split. For a 2-way BCV mirror device for a protected establish, both M1 and M2 move to the standard device mirror set and become instantly synchronized and available for updates from I/O activity on the standard device.

For example, to initiate a full `-protbcvest` establish on a 2-way BCV pair with 2-way standard `DEV001` in device group `Prod`, enter:

```
symmir -g Prod establish -full -protbcvest DEV001 BCV ld BCV005
```

For more information about the affects of `-protbcvest` during a split operation, refer to "Splitting a protectively established BCV" on page 164.

Pairing algorithm

As previously described, standard and BCV devices within a device group are selected into pairs using a full establish. For your understanding of the software behavior, the following detail the sequences of the pairing algorithm used by the full establish SYMCLI processes:

1. If a BCV device is explicitly specified, pair it with the specified standard device.
2. If the `-exact` option is specified, pair the devices in the exact order that they were added to the device group (refer to "Exact" on page 216).
3. If the optimize (`-opt`) option is specified, pair the BCV device such that it is on separate disk adapters for improved copying speed (refer to "Local optimizing" on page 217). This option overrides all other pairing decisions.
4. If the remote optimize (`-opt_rag`) option is specified for a composite group operation, pair the remote BCV device such that it is on separate disk adapters in the remote Symmetrix array, and improving copying speed (refer to "Remote optimizing option" on page 224). This option overrides all other pairing decisions.
5. If an attached BCV device exists, for the specified standard device, use the attach assignment. If the device specified was previously paired, use the previous pair assignment. If the paired BCV device is not associated with the same device

group as the standard device, you must use the force option to override the current pairing and pair the standard device with another BCV of equal size in the device group.

6. If there were no previous pair assignments, then pair standard and BCV devices that are of equal size.

Incrementally establishing BCV pairs

Incrementally establishing a BCV pair (Figure 46 on page 147) accomplishes the same thing as the establish process, with a major time-saving exception: the standard device (DEV001) copies to the BCV device (BCV001) only the new data that was updated on the standard device while the BCV pair was split. Any changed tracks on the BCV are also overwritten by the data on the corresponding tracks on the standard device.

This process is useful if the data yielded from running an application on the BCV device is not needed or if a fresh copy of current data is needed.

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

```
symmir -g DgName establish
symmir -cg CgName establish
symmir -f[ile] FileName establish
```

Note: For information on creating a device file, refer to "Device file" on page 216.

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

```
symmir -g prod establish
```

To initiate an incremental establish on a BCV pair, `DEV001`, in the `prod` device group, enter:

```
symmir -g prod establish DEV001
```

To initiate an incremental establish on a list of specific BCV pairs in the `prod` device group, enter:

```
symmir -g prod establish DEV001
                         DEV002
                         DEV005
```

The establish defaults to re-establishing the previous BCV pairing, unless you use either the `-full` option.

Figure 46 illustrates the incremental establishing of a BCV pair.

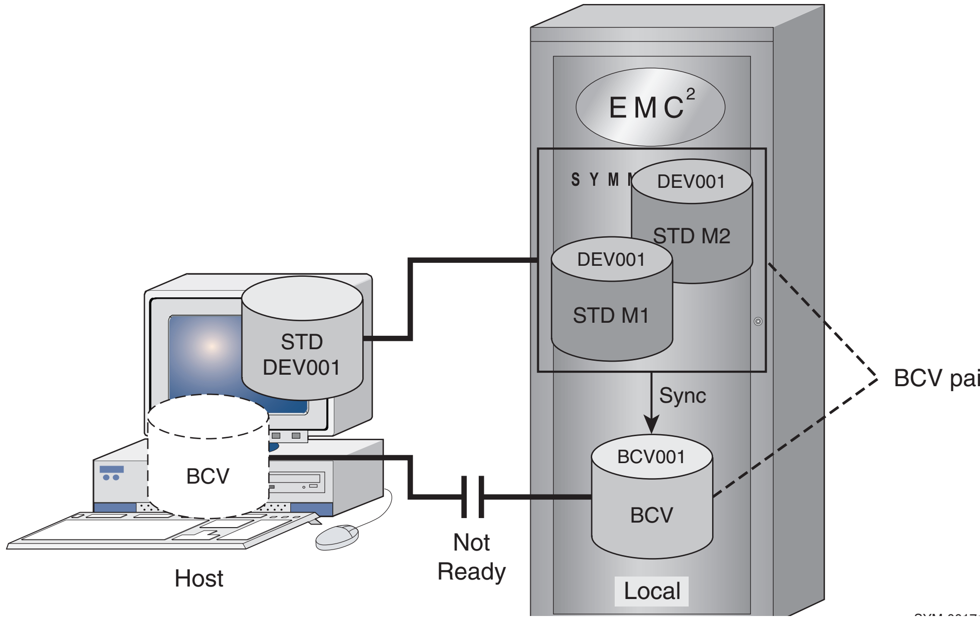

Figure 46 Incrementally establishing a BCV pair

When an incremental establish is initiated for each specified BCV pair in a device group, the following occurs:

- Command validity is checked. For example, the command is rejected if the BCV and standard devices were not previously paired.
- The BCV device is set as Not Ready to the host.
- The BCV device is assigned as the next available mirror of the standard device.
- The tracks are copied from the standard device to the BCV. Any new data written to the standard device while the BCV pair was split is written to the BCV device. Any new data written to the BCV device while the BCV pair was split is overwritten by the data on the corresponding track on the standard device.

The BCV pair is synchronized when the standard device and the BCV device contain identical data.

Note: The BCV device is not available for host use during the time it is assigned as a BCV mirror on a standard device. However, any new data written to the standard device is copied to the BCV device while the BCV pair exists.

For business continuance procedures, to make the devices usable, you must split the the devices again.

Automatically converting an incremental establish to a full establish

TimeFinder will automatically convert an incremental establish to a full establish when it determines that the requested devices (local or remote) have no prior relationship.

The SYMAPI_DEFAULT_BCV_ESTAB_INC_TO_FULL parameter in the options file allows you to enable this feature. Disabling this feature (default) will cause the SYMAPI to return the error SYMAPI_C_DEVICE_IS_NOT_PAIRED, as it did with previous versions.

Note: Enabling the SYMAPI_DEFAULT_BCV_ESTAB_INC_TO_FULL parameter does not eliminate the requirement to use the `-full` option with the `-opt` option.

Incrementally establishing multiple BCV pairs

You can incrementally establish (at different times) up to 16 (8 when using emulation) BCV devices associated with a single standard device. Note that initially, all the BCV devices must have been fully established before you perform any incremental establishes on them.

Note: Using the environment variable SYMCLI_MAX_BCV_PAIRS, the maximum number of pairs (established or restored) can be adjusted between 1 to 16 BCV devices.

With this feature, standard devices retain a relationship with multiple BCVs as long as those BCVs do not become paired with another standard. Here, the information about changed tracks is saved for a split BCV device if another BCV device is subsequently incrementally established and split from the same standard device. By invoking a series of `split/increment establish` commands over time (Figure 47), a multi-BCV environment becomes established that retains progressive historical images of the data on the specified standard.

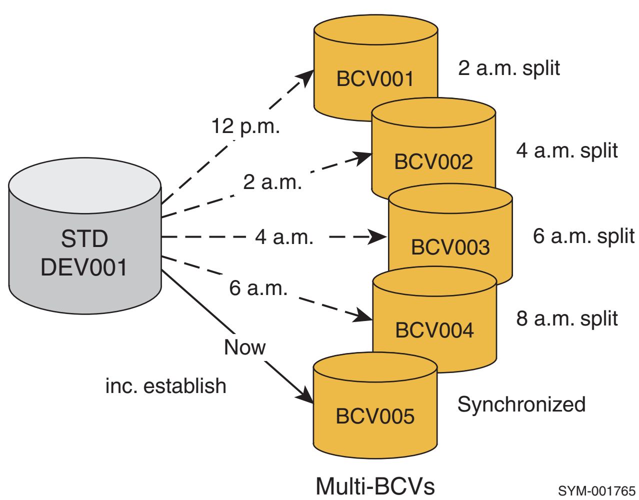

Figure 47 Establishing a multi-BCV environment

For example, to query for existing multi-BCVs that were originally all previously (full) established, and then to incrementally establish `BCV005` with the standard, enter:

```
symmir -g DgName query -multi
symmir -g DgName split DEV001
symmir -g DgName establish DEV001 BCV ld BCV005
```

In this environment, you can specify any one of these older BCVs to incrementally restore or establish the standard back to a historical copy (refer to "Incrementally restoring multiple BCV pairs" on page 182).

For instructions on cancelling the incremental relationship to one of the BCVs from the set, refer to "Canceling a multi-BCV relationship" on page 136.

Incrementally establishing SRDF-connected BCV pairs

You can specify an incremental establish action to a remote Symmetrix site using the RDF flag (`-rdf` option), which incrementally establishes the remote BCV pairs.

For example, Figure 48 illustrates an incremental establish operation in the remote Symmetrix at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf -establish
```

In this case, the flag indicates that the BCV device being established is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local RDF standard device.

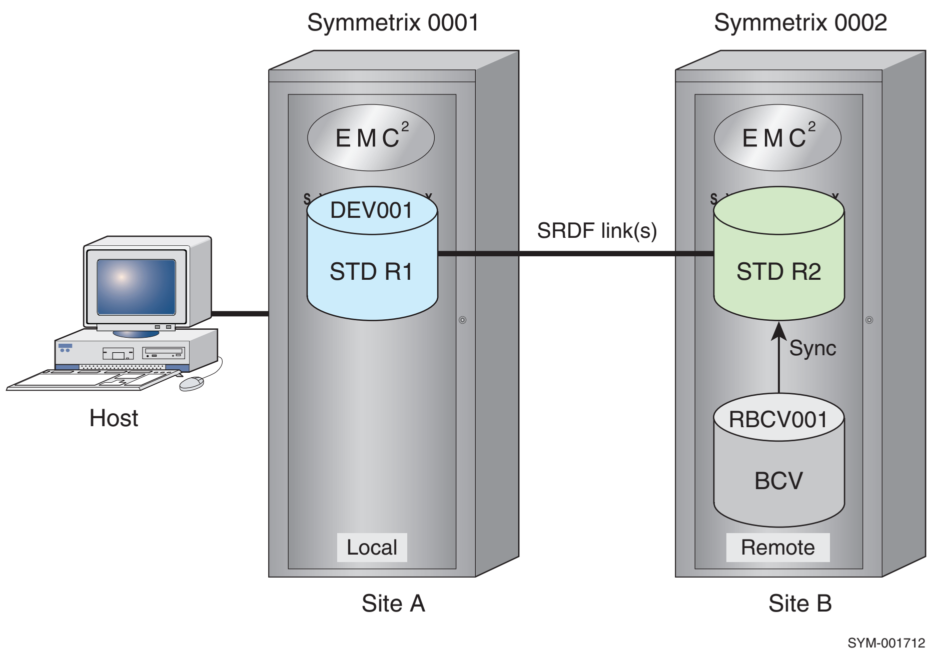

Figure 48 Incrementally establishing the remote BCV to mirror local STD

Figure 49 illustrates an incremental establish operation in the remote Symmetrix at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv establish
```

In this case, the flags indicate that the BCV device being established is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local R1 BCV device.

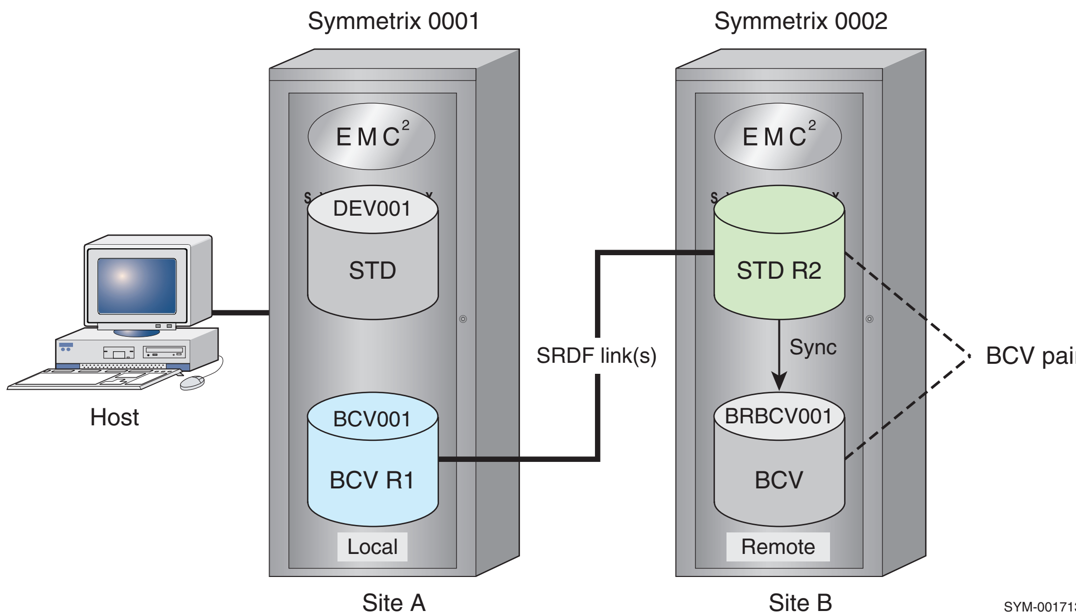

Figure 49 Incrementally establishing the remote BCV to mirror local BCV

Incrementally establishing second-level remote BCV pairs

You can specify an incremental establish action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which incrementally establishes second-level remote BCV pairs. Figure 50 illustrates a incremental establish operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv establish
```

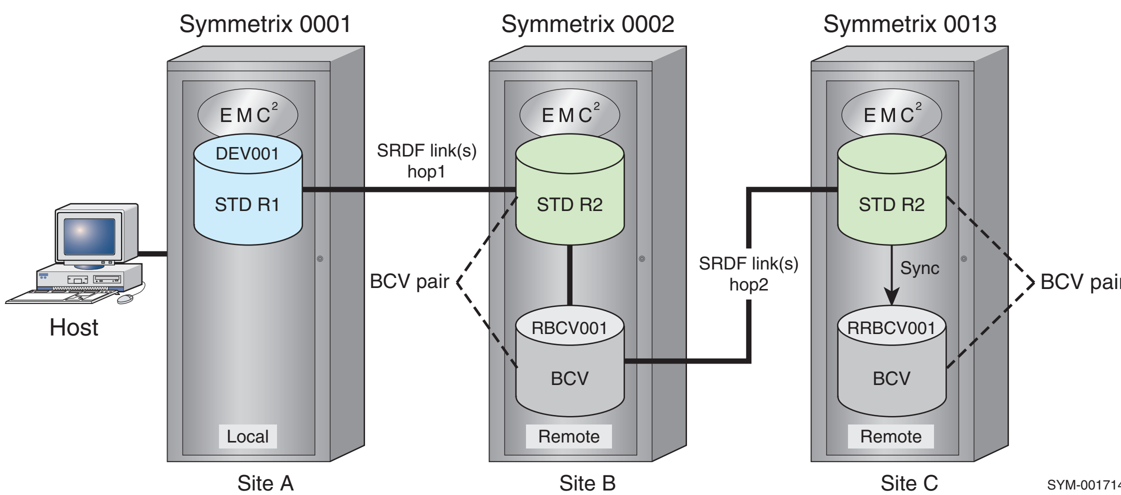

Figure 50 Incrementally establishing the remote hop 2 BCV pair (mirroring the local BCV)

To initiate a incremental establish on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv establish RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being established is a second HOP SRDF-connected BCV device, which will be established with the remote standard mirror of the remote BCV device.

Incrementally establishing hop 2 BCV pairs in a Cascaded SRDF configuration

You can specify an incremental establish action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 51 illustrates an incremental establish operation in the remote Symmetrix at the tertiary site (C) when the hop 2 flag is specified with the following command:

```
symmir -g prod -hop2 establish
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being established with the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

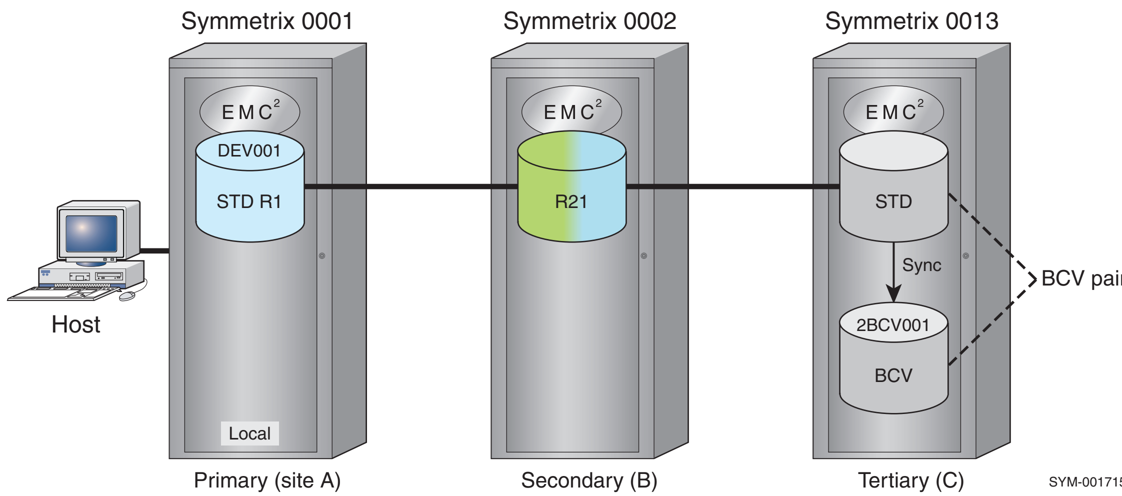

Figure 51 Incrementally establishing the hop 2 BCV pair in a Cascaded SRDF configuration

Protected BCV incremental establish (moving mirrors)

The protected BCV establish (`-protbcvest`) option (also known as the moveall establish) can be used with the incremental establish action to move all mirrors of locally mirrored BCV devices to join the mirrors of the standard device. This moving of the mirrors feature (starting with Enginuity Version 5670) improves availability of the BCV mirrors, particularly after a split operation where all mirrors are instantly synchronized.

For example, to initiate a protected (`-protbcvest)` incremental establish on a 2-way BCV pair with 2-way standard `DEV001` in device group `Prod`, enter:

```
symmir -g Prod establish -protbcvest DEV001 BCV ld BCV005
```

For more information about the protected BCV established environment, refer to "Performing a protected BCV establish (moving all mirrors)" on page 143.

Splitting BCV pairs

After an establish operation and the standard device and BCV mirrors are synchronized, the BCV device becomes a mirror copy of the standard device. You can split the paired devices to where each holds separate valid copies of the data, but will no longer remain synchronized to changes when they occur. SYMCLI processes can then be executed with the BCV device once the split completes.

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

```
symmir -g DgName split
symmir -cg CgName split
symmir -f[ile] FileName split
```

Note: For information on creating a device file, refer to "Device file" on page 216.

For example, to split all the BCV pairs in the `prod` device group, enter:

```
symmir -g prod split
```

To split a BCV pair, `DEV001`, in the `prod` device group, enter:

```
symmir -g prod split DEV001
```

To split a list of standard devices in the `prod` device group, enter:

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

Figure 52 on page 155 illustrates the splitting of a BCV pair.

Note: Relabeling devices is sometimes required in situations where devices are under an MS Windows type volume manager's control. If a BCV (TimeFinder) device holds an identical copy of its standard (paired) device, and when the BCV device becomes ready to the Windows operating system, the volume manager will detect two identical volumes with different mount points. This can cause the volume manager on Windows to exit and crash a system.

For information about device label ambiguity, refer to "Relabeling devices for Windows 2000" on page 232.

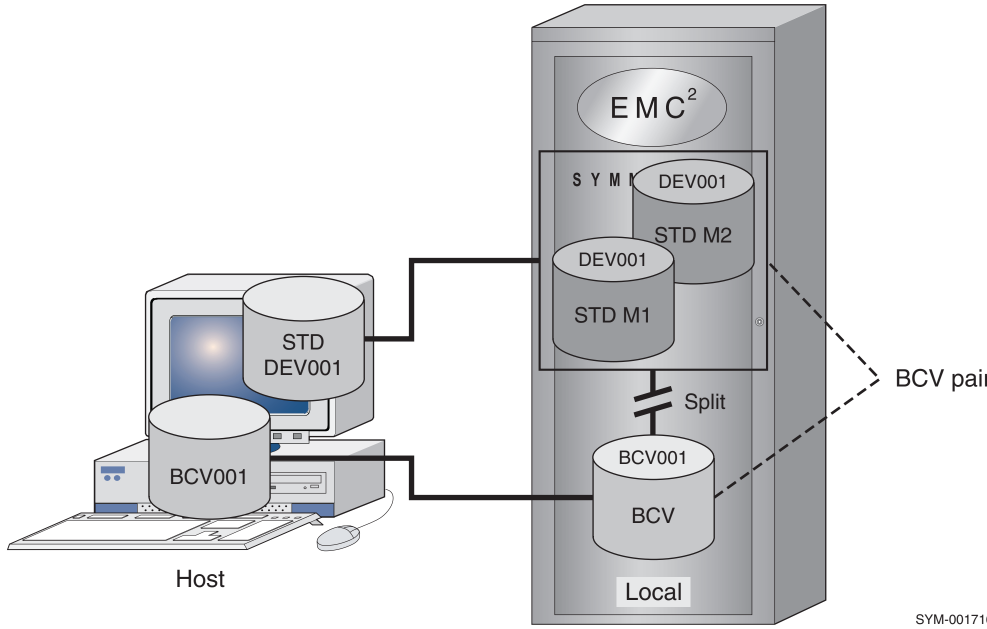

Figure 52 Split the BCV pair

When a *split* is initiated for each specified BCV pair in a device group, the following occurs:

- Command validity is checked. For example, the Symmetrix array makes sure that the standard device has an active BCV mirror and that the standard and BCV devices comprise a BCV pair.
- Any pending write transactions to the standard device and the BCV device are destaged.
- The BCV device is split from the BCV pair.
- If the device is a meta device, all meta members are implicitly split as well.
- The BCV device state is changed to Ready, enabling host access through its separate address (`BCV001`).
- Operation with the standard device is resumed and any tracks changed from write operations to the standard device are marked. (This is necessary for updating the BCV device if it is reestablished with the same standard device at a later time.)
- If the BCV device has any of its own mirrors, the mirrors are synchronized, unless protected establish or emulation is used.

Once you finish running any Business Continuance processes on the BCV device, the following options are available:

- Incremental establish or reestablish of the BCV pair.
- Establish a new pairing using the same BCV devices with a different standard device.
- Restore data to a standard device from the BCV device.
- Incrementally restore data to the standard device from the BCV device (if the devices were previously paired).

Splitting SRDF-connected BCV pairs

You can also specify a `split` action to a remote Symmetrix site using the RDF flag (`-rdf` option), which splits the remote BCV pairs.

For example, Figure 53 illustrates the splitting of a remote BCV pair using the following command:

```
symmir -g prod -rdf split
```

In this case, the flag indicates that the BCV device being split is an SRDF-connected device, which will be split from the remote standard mirror of the local RDF standard device.

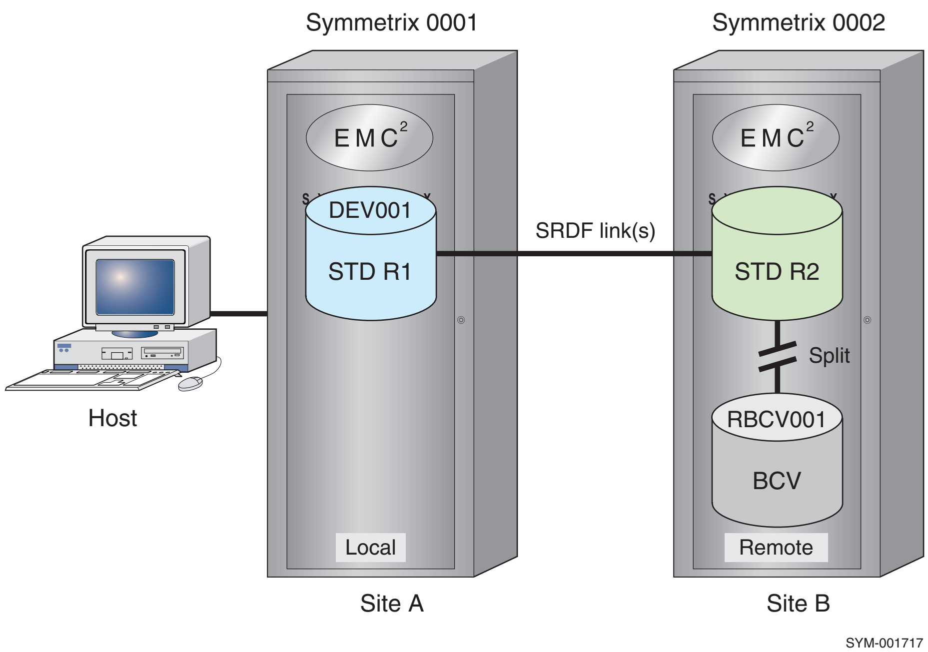

Figure 53 Split the remote BCV pair that mirrors the local STD

If a BCV device has its own mirrors (local or remote), these mirrors will become synchronized with its first mirror after the BCV pair is split.

Figure 54 illustrates a split operation in the remote Symmetrix array at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv -split
```

In this case, the flags indicate that the BCV device being split is an SRDF-connected BCV device, which will be split with the remote standard mirror of the local R1 BCV device.

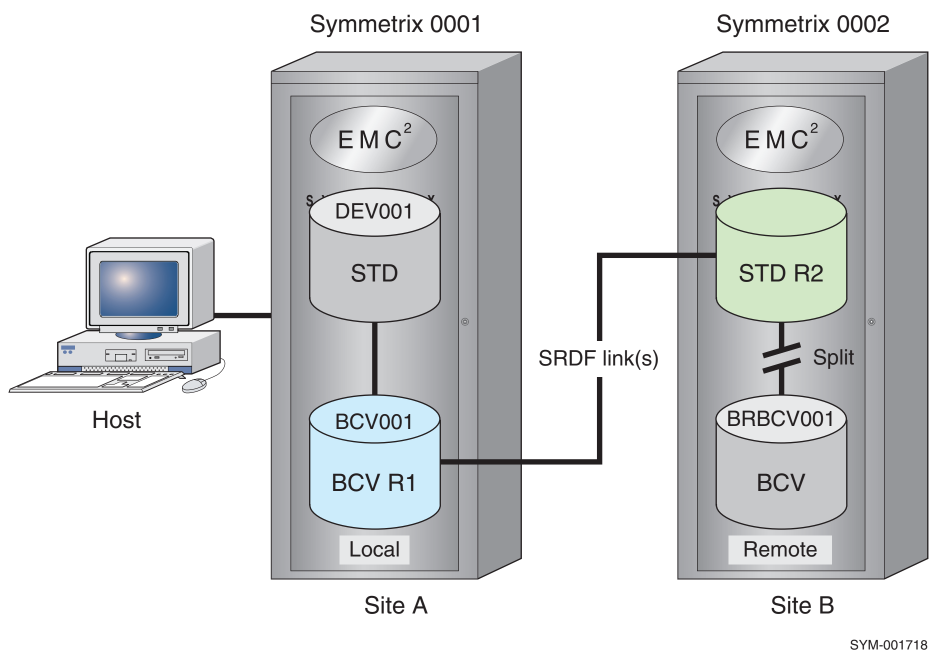

Figure 54 Split the remote BCV pair that mirrors the local BCV

Splitting second-level remote BCV pairs

You can specify a `split` action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which splits second-level remote BCV pairs. Figure 55 illustrates a split operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv split
```

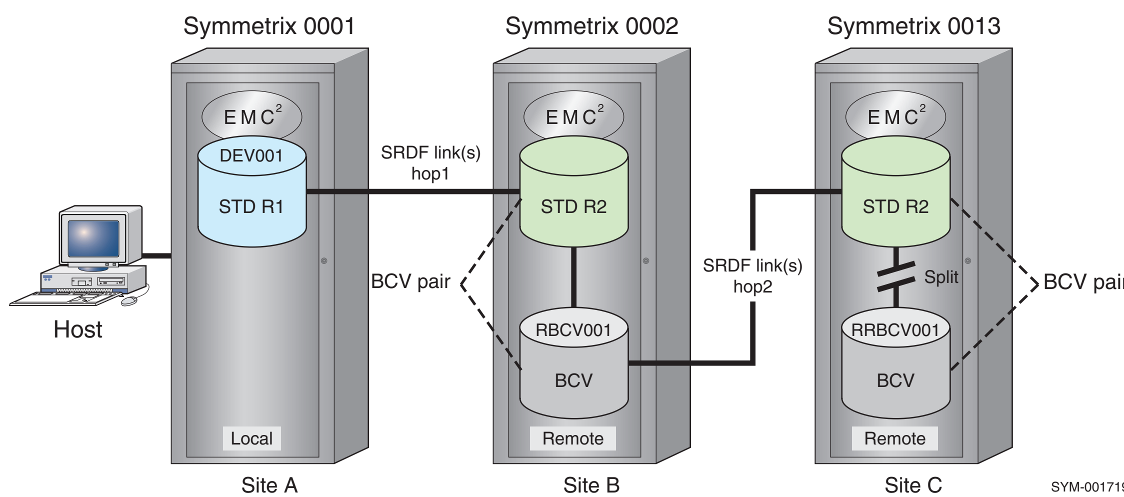

Figure 55 Splitting the remote hop 2 BCV pair

To initiate a split on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv split RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being split is a second HOP SRDF-connected BCV device, which will be split from the remote standard mirror of the remote BCV device.

Splitting hop 2 BCV pairs in a Cascaded SRDF configuration

You can specify a `split` action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 56 illustrates an establish operation in the remote Symmetrix at the tertiary site (C) when the hop 2 flag is specified with the following command:

```
symmir -g prod -hop2 split
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being split from the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

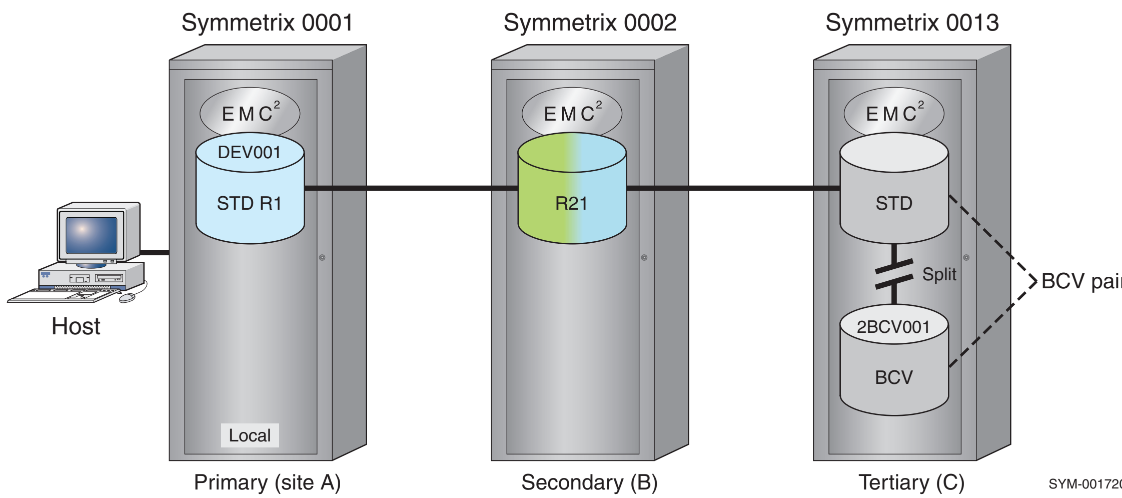

Figure 56 Splitting the hop 2 BCV pair in a Cascaded SRDF configuration

Performing a remote copy with BCV split

In addition to splitting a local BCV pair, you can further specify the remote (`-remote`) option which make the R1 device ready on the link and propagate the R1 BCV copy across the SRDF link to the R2 that mirrors the BCV:

```
symmir -g prod split DEV001 -remote
```

Figure 57 illustrates the remote copy to the R2 mirror from the local BCV split.

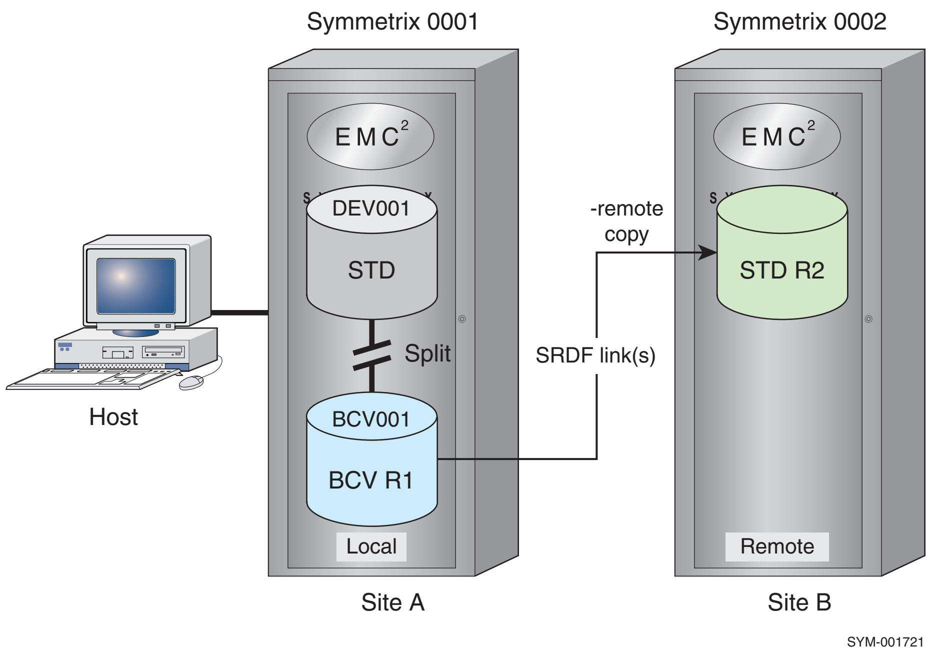

Figure 57 Remote copy with a BCV split

Performing a Protected split

The standard protect (`-std_protect`) option verifies that the standard device has either mirror protection, or RAID-5 protection prior to allowing the split to take place.

To verify a standard device's mirror protection before allowing the split operation, enter:

```
symmir -g prod -std_protect split
```

If the standard device will be left in an unprotected state after the split, or if there are any invalid tracks on the standard device's static mirrors, the command will fail.

Note: This functionality is not supported in TimeFinder/Clone Emulation mode.

Performing a reverse split

The reverse (`-reverse`) option initiates a reverse data copy from the fixed BCV mirror to the primary (moving) mirror of the BCV upon the completion of the split operation. The reverse split feature is supported in Enginuity Version 5x68 and higher.

Note: This functionality is not supported in TimeFinder/Clone Emulation mode.

Normal split behavior

Normally, when a BCV has two mirrors, only the primary mirror (M1) joins the standard device in establish or restore operations. As shown in Figure 58, the content of the primary BCV mirror is refreshed by data from the standard, when the BCV is established. The secondary BCV mirror (M2) is refreshed by data from the primary BCV mirror (M1), after the BCV mirror is split from the standard. The primary BCV mirror is referred to as the *moving mirror*, because it moves between the standard and the secondary (fixed) mirror.

Usually, after a split, the fixed BCV mirror is refreshed from the moving BCV mirror. This can either be a full copy operation or a differential copy. In a differential copy, only the tracks that have changed on the moving mirror during the time it was synchronized with the standard are refreshed.

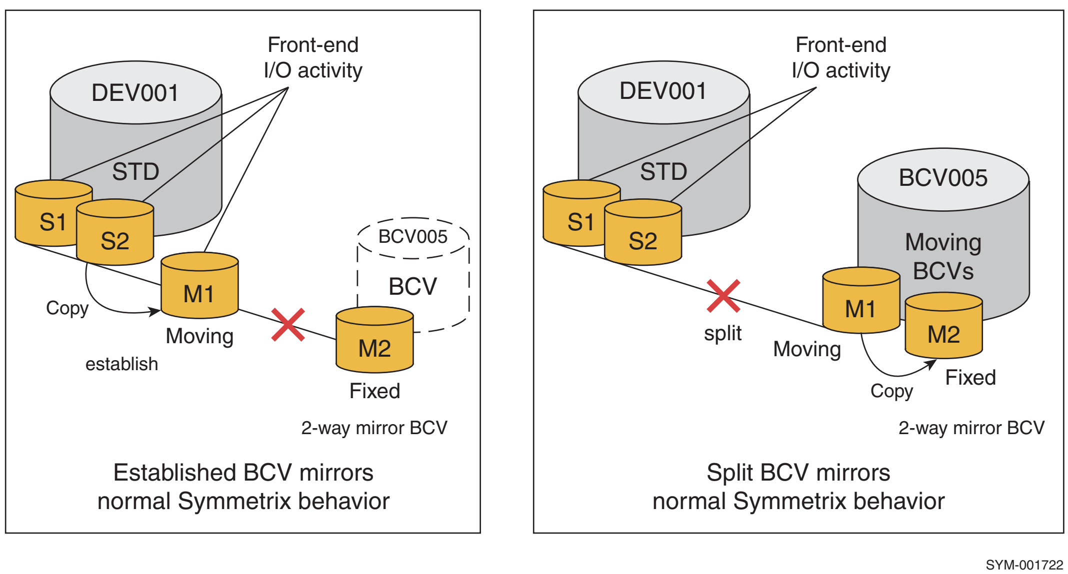

Figure 58 Two-way mirror BCV establish/split normal behavior

Reverse split behavior

In a reverse split operation, the direction of data flow between the BCV mirrors is reversed. As shown in Figure 59, during a reverse split, the fixed BCV mirror (M2) will refresh the moving mirror (M1) after the split operation. This behavior may be desirable when you need to revert to an older copy of the data that was on the BCV before it was established/restored with the standard.

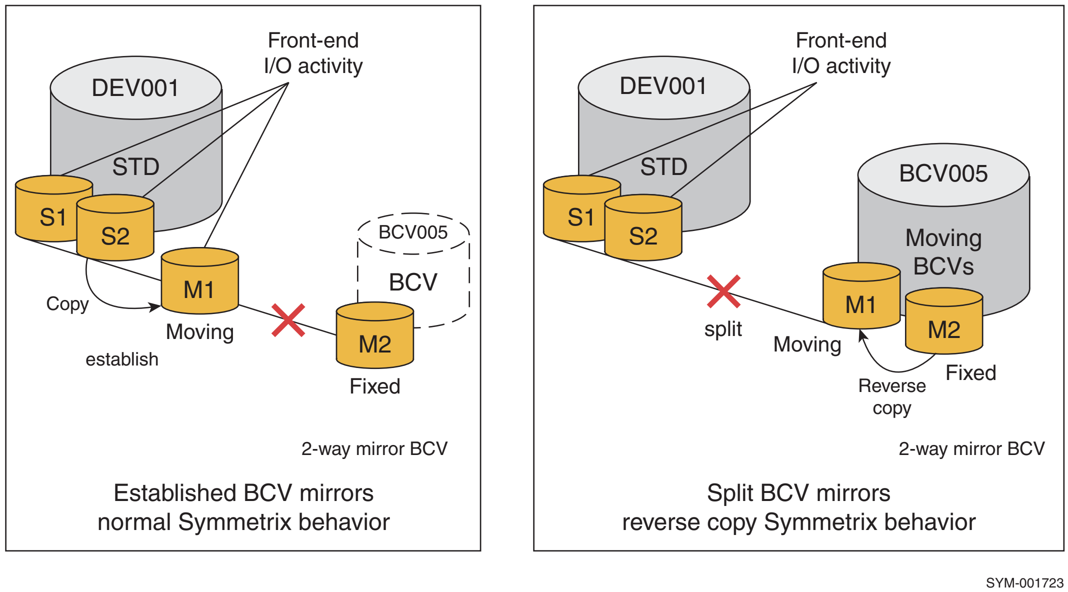

Figure 59 Two-way mirror reverse establish/split behavior

CAUTION

Be sure this is the behavior you want before invoking the reverse split option since the primary BCV mirror data is refreshed with an older mirror of data.

A reverse split is permitted only if both BCV mirrors were completely synchronized before the moving BCV mirror was paired with the standard device. When you anticipate a need for future reverse split operations, the `-reverse` option is applied to an establish or restore operation. This option requests a verification check that the BCV's fixed mirror has valid data. You must verify that both mirrors are in the Ready state after the split.

For example, to establish `DEV001/BCV005` and later perform a reverse split on `DEV001` in device group `Prod`, enter:

```
symmir -g Prod establish -reverse DEV001
symmir -g Prod split -reverse DEV001
symmir -g Prod verify -bcv_mirrors -ready
```

Note: Data may not be immediately available. Use the `verify` command to check that both mirrors are in the Ready state for the data to be available.

Figure 60 illustrates a practical use of a reverse split. At midnight a split results in a good point-in-time copy of a database. At 10 A.M. a corruption is discovered in the database, necessitating a database recovery. At 10:30 A.M. a restore operation is initiated from the BCV copy. Because the good data is immediately available to the BCV pair, the recovery begins shortly after initiating the restore process. At 11:00 A.M, during the recovery, one of the logs re-corrupts the database. Though the data on the BCV's moving mirror has changed during the recovery process, a reverse split can be initiated. At 11:20 A.M. the BCV's fixed mirror refreshes its moving mirror, providing access to the good midnight copy of the data.

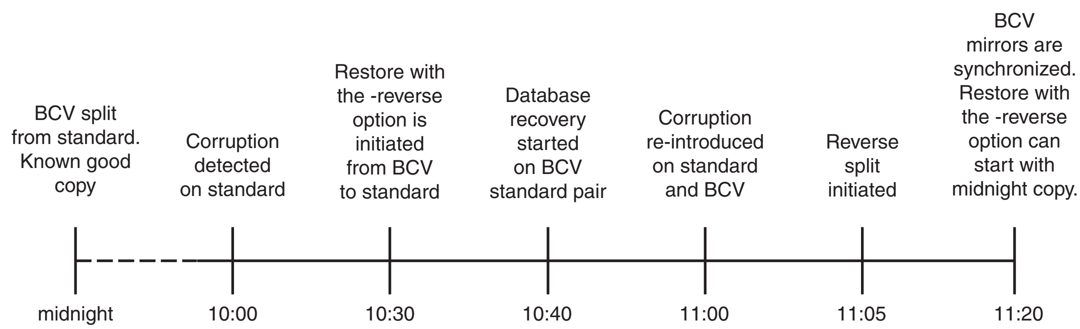

Figure 60 Practical use of a reverse split

Splitting concurrent BCV pairs

Since Enginuity Version 5x68, you can establish two BCV devices as concurrent mirrors of a single standard device. This relationship is known as a *concurrent* BCV pair. This feature allows you or an application to instantly generate two synchronized copies of the standard data (refer to "Establishing concurrent BCV pairs" on page 142).

When you apply a split action to a standard device that was concurrently established with two BCV mirrors, both BCVs become split from the standard. Moreover, the splitting of concurrent BCVs will be *instantly* split without explicitly invoking the `-instant` option.

For example, to split concurrently established `BCV001` and `BCV002` with standard `DEV012` in device group `CncGrp`, enter:

```
symmir -g CncGrp split DEV012
```

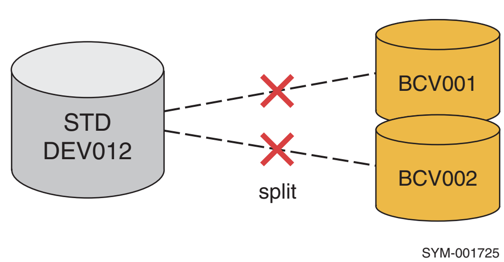

Figure 61 Splitting concurrent BCV pairs

Individual split of concurrent BCVs

If you do not want both of the concurrent BCVs to simultaneously split together, you can individually target the split action by explicitly specifying the BCV with the standard as follows:

```
symmir -g CncGrp split DEV012 bcv ld BCV001
```

or:

```
symmir -g CncGrp split DEV012 bcv ld BCV002
```

Checking concurrent splits

To display the status of the background concurrent split for both of these BCVs, enter:

```
symmir -g CncGrp query -multi -bg
```

Because invalid track tables are maintained, future concurrent incremental establish operations are possible on these split BCVs. After a concurrent split, it is possible to resynchronize just one BCV as follows:

```
symmir -g CncGrp establish DEV012
```

However, if you do not explicitly specify a BCV and you have not set an attachment preference, TimeFinder chooses the BCV to resynchronize, based on which of the two concurrent BCVs was split first.

You can also verify that the action has completed, as follows:

```
symmir -g CncGrp -split verify -bg DEV012 BCV ld BCV002
```

Splitting a protectively established BCV

A split action splits both of the BCV mirrors away from the standard device that were previously established with a `-protbcvest` option. Figure 62 illustrates the initial protected BCV established state and the resulting behavior of a split action on these BCV mirrors. For any split command, there is no need to apply the `-protbcvest` option to move all the mirrors away from the standard.

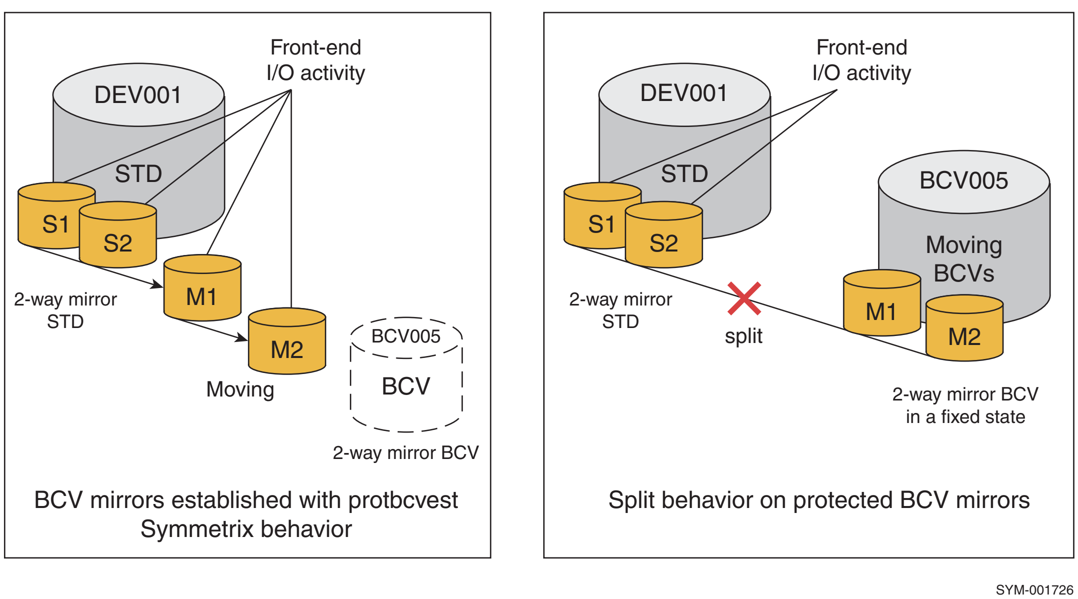

Figure 62 Split behavior on two-way BCV mirrors

Before you split, you may need to query the Symmetrix array to examine the protected STD/BCV mirrored environment to identify the established moved devices for the split action, as follows:

```
symmir -g prod query -protbcvest
```

To perform a protected establish and later split (for example) on standard DEV001 with its BCV mirrors, in device group prod, enter:

```
symmir -g prod establish -protbcvest DEV001
   .
   .
   .
symmir -g prod split DEV001
```

For more information about a protected BCV establish, refer to "Performing a protected BCV establish (moving all mirrors)" on page 143.

BCV splits impacting database applications

Occasionally, a BCV split can impact the integrity (ability to restart) of a database on the split copy. In such a case, additional actions such as freezing the database to user access may be necessary. The freeze action can be used in conjunction with the TimeFinder split operation. The freeze suspends the database updates being written to disk.

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

Note: First, you must set SYMCLI_RDB_CONNECT to your username and password for access to the specified database. For more information, refer to the symioctl command description in the *Solutions Enabler Symmetrix CLI Command Reference*.

Freeze

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

```
symioctl freeze -type DbType Object Object
```

SQL Server allows some or all databases to be specified. Oracle and Informix let you freeze or thaw an entire DB system.

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

```
symioctl freeze Object Object
```

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

```
symioctl freeze HR Payroll
```

Thaw

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

```
symioctl thaw
```

Hot backup control

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

1. Issue the symioctl begin backup command.
2. Invoke symioctl freeze.
3. Split standard and BCV pairs. (This may involve several steps depending on your environment.)
4. Invoke symioctl thaw.

5. Issue the `symioctl end backup` command.

This operation on an Oracle database does not directly affect meta data I/O. The operation does not affect any Oracle files that are not related to this database instance or any non-Oracle files in the same file system. Consequently, just a freeze and thaw action does not guarantee an Oracle database integrity as consistent split does. However, this operation does enhance the performance of splits if you apply a consistent TimeFinder split operation.

Inconsistent data problems on splits

In database environments (Figure 63) when multiple Symmetrix devices are being split at the same moment, typical TimeFinder split operations are performed in an unpredictable order due to the unpredictable nature of I/O load on each device. If a database is split without quiescing, it is possible to end up with BCVs holding user data that is more recent than the BCVs holding log data. This leads to a BCV image of the database that is inconsistent and not restartable.

In addition to log-older-than-data problems, database *fuzziness* and *fractured blocks* can occur. Fuzziness is the result of each STD/BCV pair not being split at exactly the same moment in time. Fractured blocks can occur if a pair splits in the middle of an Oracle-specific I/O.

Previously, to overcome these inconsistent split problems, you had to place your database in backup mode or freeze I/O to the database (such as Informix, SQL Server, etc.) before issuing a BCV split. This, in turn, gave you a restartable database image on the BCVs that could be backed up or started on a backup server. But now, a TimeFinder *consistent split* circumvents this tedious operation and is a cleaner and faster split on the BCV database environment. For more information, refer to "TimeFinder consistent split" on page 167.

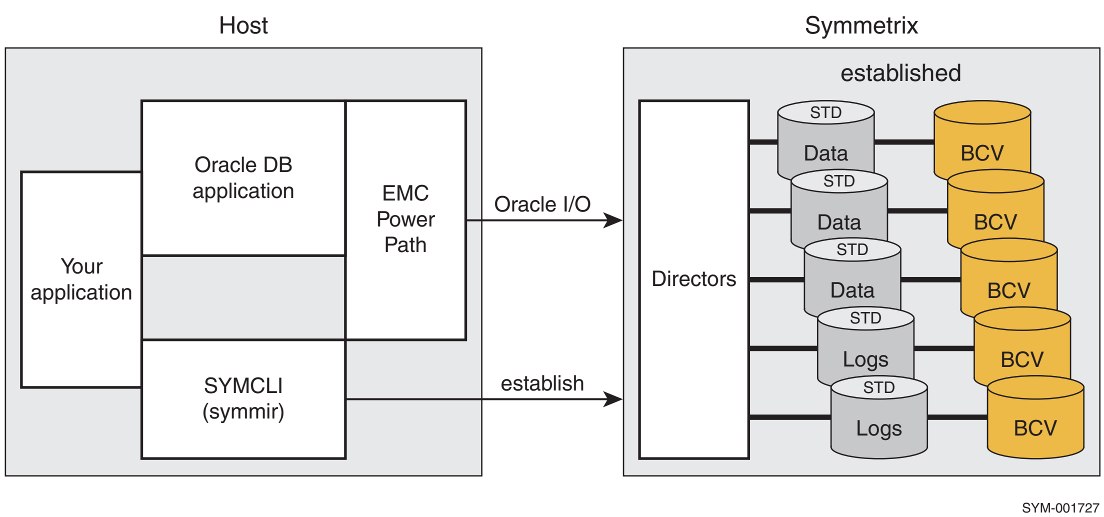

Figure 63 An established Oracle DB environment

TimeFinder consistent split

TimeFinder consistent split allows you to split off a consistent, restartable copy of a database management system within seconds with no interruption to online service. Consistency split helps to avoid inconsistencies and restart problems that can occur when splitting a database-related BCV without first quiescing the database. Consistent split can be implemented using either PowerPath®-connected devices, Enginuity Consistency Assist (ECA) functionality, or SRDF/A.

Note: This functionality requires a TimeFinder/CG license. "Consistent split using Enginuity Consistency Assist" on page 169 contains greater detail. The *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*. provides complete details on SRDF/A.

Consistent split using PowerPath

Consistent splits using PowerPath can be implemented with an instant (`-instant`) split where you must also specify either a database or PowerPath device(s) and any pre-action and post-action scripts:

To target a database, use the following syntax:

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

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

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

PowerPath consistent split operations require Version 2.0.1 or higher PowerPath-connected devices on Symmetrix arrays. Through the assistance of PowerPath, a `symmir` consistent split (supplied with database or PowerPath parameters) initially suspends I/O to the devices that hold the database. This prevents the database application from proceeding.

The consistency split command also lets you specify the name of scripts using the `-preaction` and `-postaction` script options. The script commands are executed prior to the freeze and after the thaw operation, respectively.

Coding pre- or post-action scripts

When you code pre-action or post-action scripts for use with the `symmir` instant split command, you must return to the execution sequences of the `symmir` command with a `0` (zero for success) or a nonzero value (for any failure type unique to your script).

It is recommended you use the `symreturn` command supplied with SYMCLI for this purpose to exit your pre- or post-action scripts and re-enter the SYMCLI `symmir` instant split processing. Again, with `symreturn`, you supply a `0` argument for success or any nonzero value for a failure.

For example, to return success, enter in your script:

```
symreturn 0
```

Figure 64 shows the sequences to execute a consistent instant split operation on an Oracle environment.

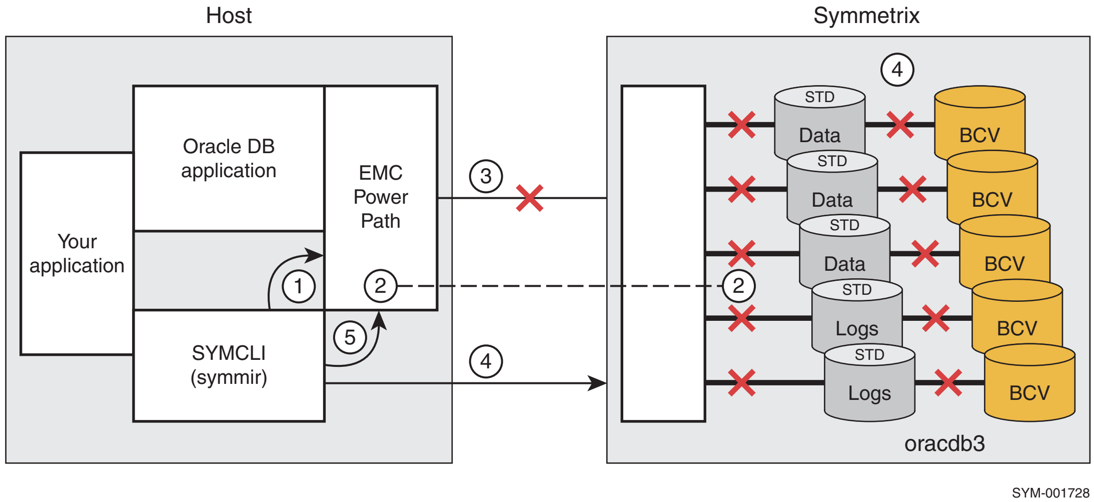

Figure 64 Consistency split on Oracle environment

For example in the figure, the consistent instant split sequence starts with:

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

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

Consistent split on both RDF sides using PowerPath

In an RDF environment as shown in Figure 65, you can perform a consistent split to the BCVs in both the local Symmetrix array and the remote Symmetrix array. This functionality requires a TimeFinder/CG license.

Note: This feature is not supported with R21 groups

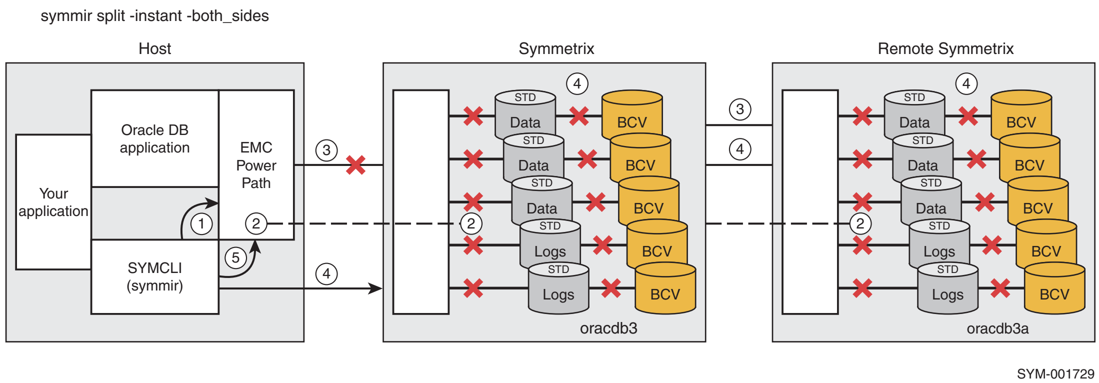

Figure 65 Consistent split on both RDF sides using PowerPath

The consistent instant split on both RDF sides is only implemented with an instant (`-instant`) split where you must also specify a database or PowerPath device(s).

To target a database target (both sides), use the following syntax:

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

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

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

Consistent split using Enginuity Consistency Assist

You can use Enginuity Consistency Assist (ECA) to perform consistent split operations across multiple, heterogeneous hosts without the use of PowerPath support. This functionality requires a TimeFinder/CG license.

Using ECA to consistently split BCV devices from the standards, you must have either a control host with no database or a database host with a dedicated channel to the gatekeeper devices. The dedicated channel cannot be used for servicing other devices to freeze I/O. Refer to Figure 66 on page 170 for a depiction of how a control host can perform ECA consistent splits for three database hosts that access devices on a Symmetrix array.

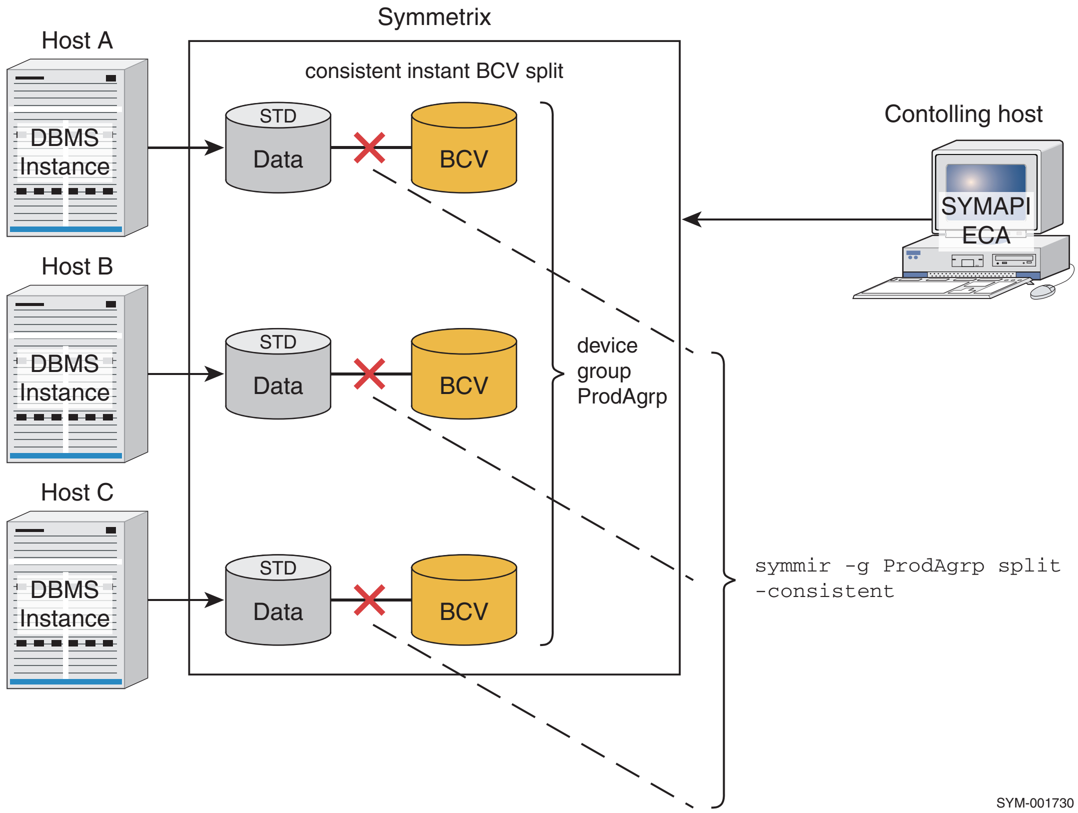

Figure 66 ECA consistent split

Symmetrix device groups or composite groups must be created on the controlling host for the target database to be consistently split. Device groups can be created to include all of the devices being accessed or defined by database host access. For example, if you define a device group that includes all of the devices being accessed by Hosts A, B, and C (refer to Figure 66), then you can consistently split all of the BCV pairs related to those hosts with a single command. However, if you define a device group that includes only the devices accessed by Host A, then you can split those BCV pairs related to Host A without affecting the other hosts.

Note: For information on performing a consistent split to BCVs in both the local and remote Symmetrix arrays, refer to "Consistent split on both RDF sides using ECA" on page 172.

The following steps explain the example in Figure 66 of how to create one device group including all database host accessed devices and perform a consistent split operation on all of the BCV pairs accessed by those hosts.

1. Create a REGULAR type device group:

```
symdg create ProdAgrp -type REGULAR
```

2. Add all of the standard devices holding the database for each host (A, B, and C) to the device group:

```
symld -g ProdAgrp addall dev -range 0286:028B
symld -g ProdAgrp addall dev -range 0266:026B
symld -g ProdAgrp addall dev -range 0246:024B
```

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

```
symbcv -g ProdAgrp associateall dev -range 039A:039F
symbcv -g ProdAgrp associateall dev -range 037A:037F
symbcv -g ProdAgrp associateall dev -range 035A:035F
```

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

```
symmir -g ProdAgrp establish -full -noprompt
```

Note: When the BCV pairs in device group ProdAgrp (Host A, B, and C - BCV pairs) are synchronized, you can perform the consistent split using the `symmir split` command to split all of the BCV pairs associated with those hosts.

5. Use the `-consistent` option to perform a consistent (instant) split on all BCV pairs in the device groups:

```
symmir -g ProdAgrp split -consistent
```

Once the `symmir split -consistent` command is issued, I/O to the device group is frozen and a 30-second Enginuity protection timer begins. After the split completes (or 30 seconds passes, whichever comes first), the I/O channels are thawed, granting (pent up) operations access to the standard devices. Splits across all devices in a group are considered consistent, if the BCV split execution is performed within that window.

If for some unknown host or I/O channel reason, not all devices are split within the 30-second window, `symmir` returns the following reply at completion:

```
Consistency window was closed on some devices before the operation completed.
```

At this point, the final successful split outside the window is no longer considered to be consistent in execution across the device group. For consistency and reliability sake, it is recommended that you reestablish the device group, and then (later) attempt the consistent split again.

Table 13 provides a comparison of support criteria characteristics for using Enginuity Consistency Assist (ECA) versus PowerPath.

Table 13 Consistent split comparison: ECA versus PowerPath

Support criteria	-consistent	-ppath
Multihost support	Yes	No
Available for non-PowerPath operating systems	Yes	No
Control host capability from a non-DBMS host	Yes	No
Requires a non-database control host or dedicated channel host	Yes	No

Consistent split for SRDF/A devices

TimeFinder consistent split allows you to consistently split BCVs from R2 devices operating in asynchronous mode (SRDF/A).[1]

Although not required for SRDF/A mode, it is recommended that you use TimeFinder BCVs at the remote site to mirror R2 devices and preserve a consistent image of data before resynchronization operations. Also, R2 device BCVs can be split off of the R2 without having to drop the RDF links and without disruption to the

1. SRDF/A requires Enginuity Version 5670 or later.

SRDF/A operational cycles. R2 BCVs can be controlled from the R1-side or the R2-side host as long as the device groups have been defined on that host. Controlling the R2 BCVs from the R1-side host requires using the `symmir` command with the `-rdf` option. For example, to consistently split BCVs off the R2 RDF/A device in group `prod` from the R1 host, enter:

```
symmir -g prod split -rdf -consistent
```

Consistent split on both RDF sides using ECA

In an RDF environment as shown in Figure 67, you can perform a consistent split to the BCVs in both the local and remote Symmetrix arrays. This functionality requires a TimeFinder/CG license.

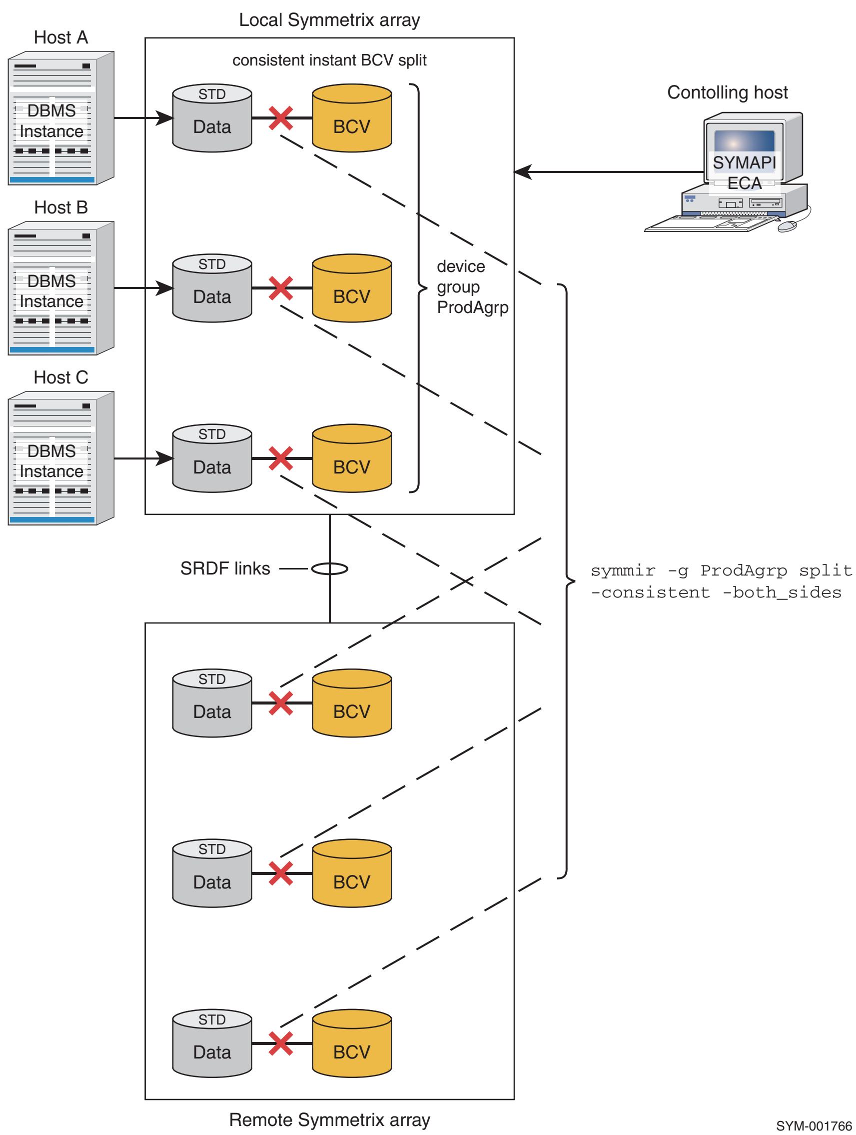

Figure 67 Consistent splits on both SRDF sides using ECA

In the above example, the controlling host issues a single consistent split command with the `-both_sides` option to split devices on both the local and remote Symmetrix arrays. For the host to perform this operation, the SRDF links must be up, the RDF mode must be synchronous, and the devices must have an RDF state of Synchronized.

Fully restoring BCV pairs

Like the full establish operation, a full restore operation copies the entire contents of the BCV devices to the standard device.

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

```
symmir -g DgName -full restore
symmir -cg CgName -full restore
symmir -f[ile] FileName -full restore
```

Note: For information on creating a device file, refer to "Device file" on page 216.

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

```
symmir -g prod -full restore
```

To initiate a full restore on a BCV pair, `DEV001`, in the `prod` device group, enter:

```
symmir -g prod -full restore DEV001
```

To initiate a full restore on more than one (list) of BCV pairs in the `prod` device group, enter:

```
symmir -g prod -full restore DEV001 BCV ld BCV001
                             DEV002 BCV ld BCV002
                             DEV005 BCV ld BCV003
```

The restoration process (Figure 68) is complete when the standard device and BCV device contain identical data.

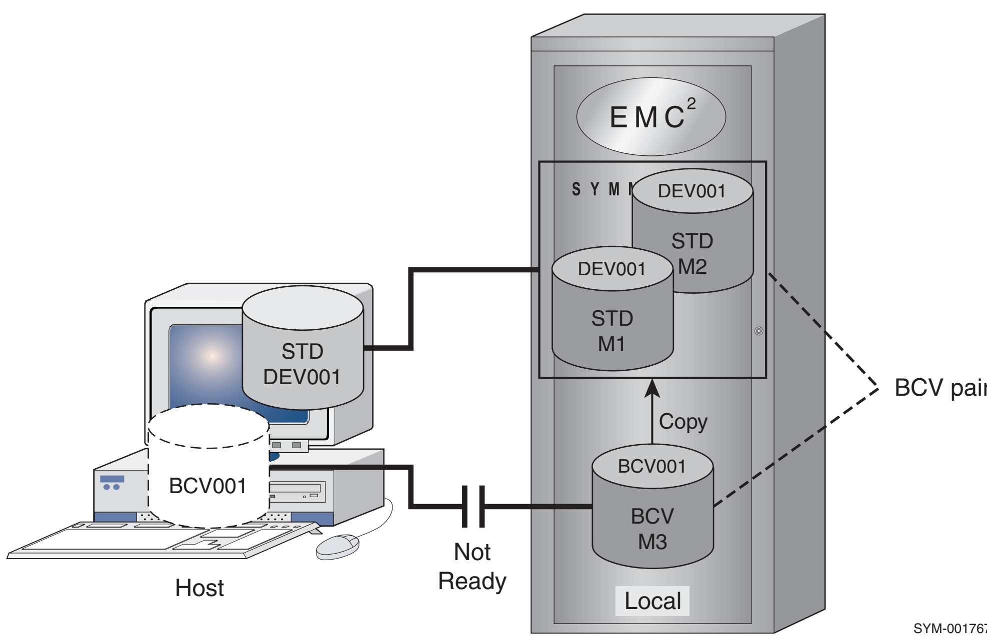

Figure 68 Full restore of the BCV pair

Note: The BCV device is not available for host use during the time that it is assigned as a BCV mirror on a standard device. However, unless the `-protect` option is used, any new data written to the standard device is copied to the BCV device while the BCV pair exists.

When a full restore is initiated for each specified BCV pair in a device group:

- Command validity is checked. For example, the command is rejected if the BCV device and the standard device are not the same size.
- The BCV device is set as Not Ready to the host.
- The BCV is assigned as the next available mirror of the standard device.
- The contents of the BCV device are copied to the standard device. For example, in Figure 68 on page 173, the Symmetrix array copies the contents of M3 to both M1 and M2, overwriting the data present on those devices.

To use a BCV device for Business Continuance procedures, you must again split the BCV pair to make the BCV device available to its host. If you want to use a fully synchronized copy of the data, suspend all applications that are using the standard device, and make sure that all host buffering and intermediate caching is flushed to the appropriate device on the Symmetrix array prior to performing the split operation. If you do not require a synchronized copy of the data for running a Business Continuance process, this step is unnecessary.

Note that the base tasks performed with `symbcv` such as list, associate, and disassociate locally or remotely connected BCV devices, are described at the beginning of this chapter.

Specifying the default method for restoring BCV pairs

The SYMAPI_DEFAULT_BCV_RESTORE_TYPE parameter in the options file enables you to specify the default method for restoring BCV pairs. Valid values are:

- SINGULAR specifies to issue the restore to one device at a time. This method allows other tasks access to the Symmetrix array when doing a large number of restores.
- PARALLEL (default) specifies to issue the restore to each servicing DA in parallel, and then wait for a DA to finish before issuing another restore to that DA.
- SERIAL specifies to issue restores as fast as the GST queue can handle them. However, all members of a meta must be restored before continuing to the next meta or device. This is the default method when using meta devices.

Note: The *EMC Solutions Enabler Installation Guide* contains information on changing the option file parameters.

Instantly restoring multiple BCV pairs

The Multi/Instant Restore option improves the performance of a typical restore operation by submitting multiple BCV pairs in a single system call to be restored instantly. This feature requires Enginuity Version 5671 or greater and will be ignored on previous Enginuity versions.

You can enable (default)/disable this feature with the SYMAPI_TF_MULTI_ESTAB_REST environment variable. Setting the SYMAPI_DEFAULT_BCV_RESTORE_TYPE to SERIAL or SINGULAR will cause this option to be ignored.

Restoring SRDF-connected BCV pairs

You can also specify a full `restore` action to a remote Symmetrix site using the RDF flag (`-rdf` option), which fully restores the remote BCV pairs.

For example, Figure 69 illustrates a full restore operation in the remote Symmetrix at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf -full restore DEV001
```

In this case, the flag indicates that the BCV device being restored is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local RDF standard device.

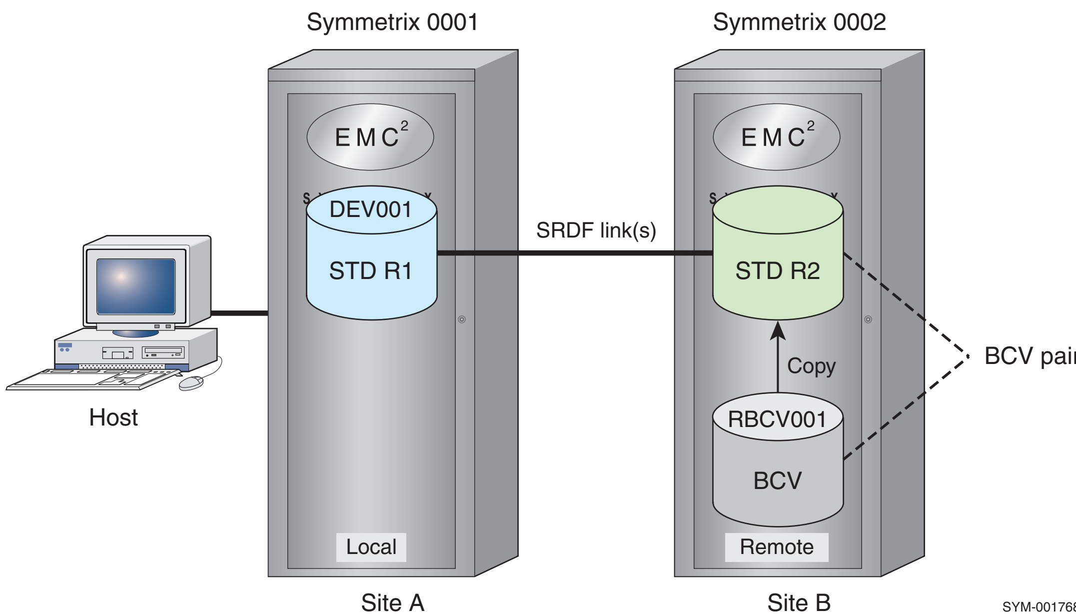

Figure 69 Full restore of remote STD that mirrors local STD

Figure 70 illustrates a full restore operation in the remote Symmetrix at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv -full restore BCV001
```

In this case, the flags indicate that the BCV device being restored is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local R1 BCV device.

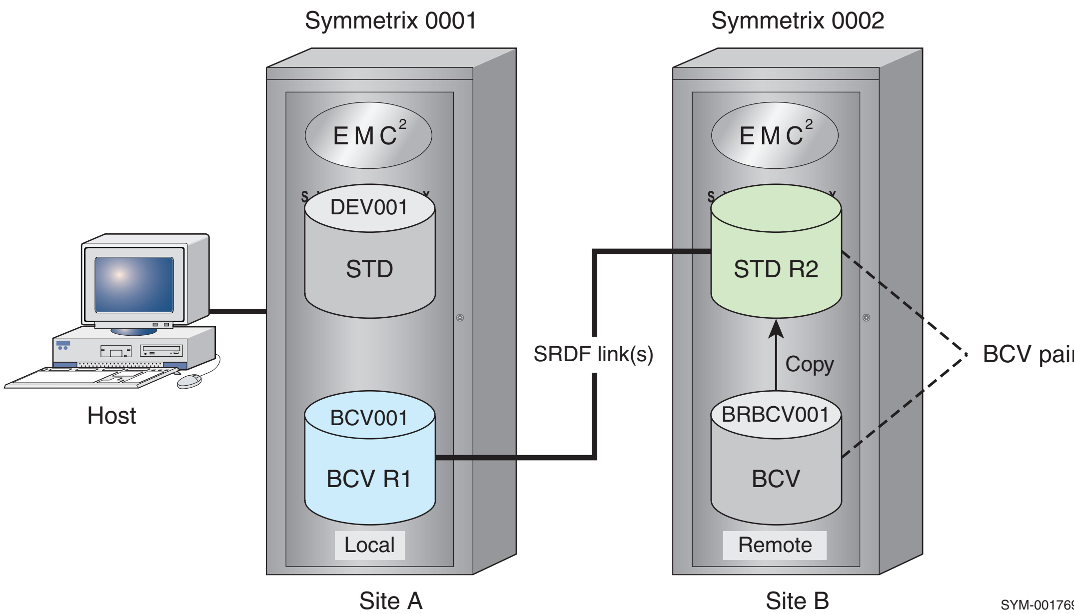

Figure 70 Restore of remote STD that mirrors local BCV

Restoring second-level remote BCV pairs

You can specify a full restore action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which restores second-level remote BCV pairs. Figure 71 illustrates a restore operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv restore -full
```

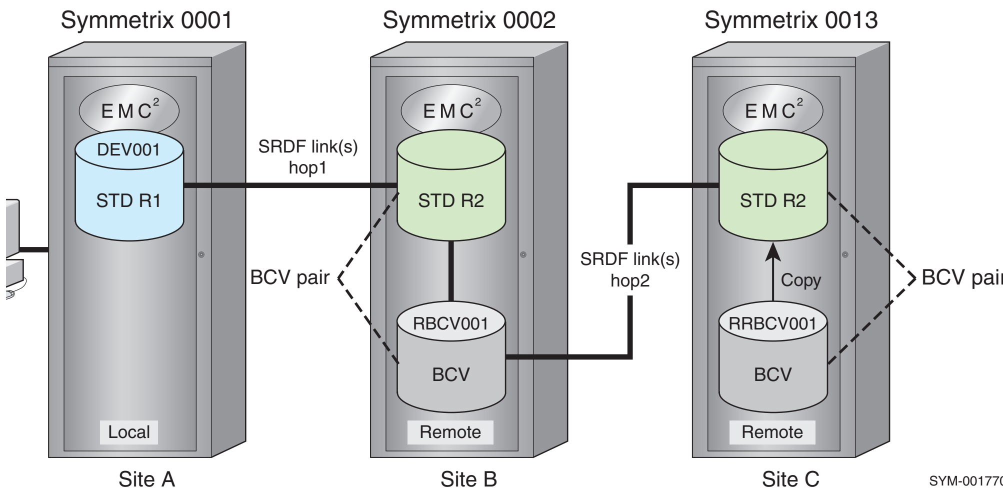

Figure 71 Restoring the remote hop 2 BCV pair

To initiate a restore on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv restore RBCV001 BCV ld RRBCV001 -full
```

In this case, the flag indicates that the BCV device being restored is a second HOP SRDF-connected BCV device, which will be established with the remote standard mirror of the remote BCV device.

Restoring hop 2 BCV pairs in a Cascaded SRDF configuration

You can specify a full `restore` action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 72 illustrates a restore operation in the remote Symmetrix array at the tertiary site (C) when the hop 2 flag (`-hop2` option) is specified with the following command:

```
symmir -g prod -hop2 restore -full
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being established with the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

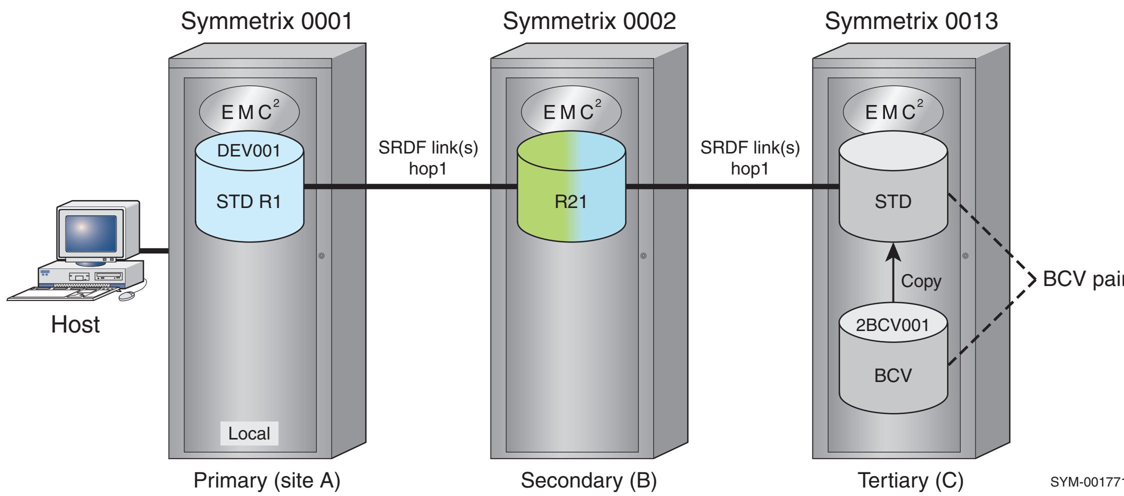

Figure 72 Restoring the hop 2 standard in a Cascaded SRDF configuration

Performing a remote copy with restore

In addition to restoring the specified BCV pair, you can further specify the remote (`-remote`) option, which will propagate the restored copy from the BCV pair across the SRDF link from the R2 standard to its R1 BCV in one command step:

```
symmir -g prod restore -rdf -bcv -remote -full
```

Note: Since Enginuity Version 5568, the `-remote` option is no longer supported for restoring from an R2 standard to an R1 BCV. Instead you should use the two step method illustrated in Figure 73 on page 179.

Note: The `-remote` option (illustrated in Figure 74 on page 179) is required when restoring to an R1 device that is ready on the link.

CAUTION

Be sure you want the R1 BCV device to be updated with the restored copy before issuing the `-remote` option along with this restore operation.

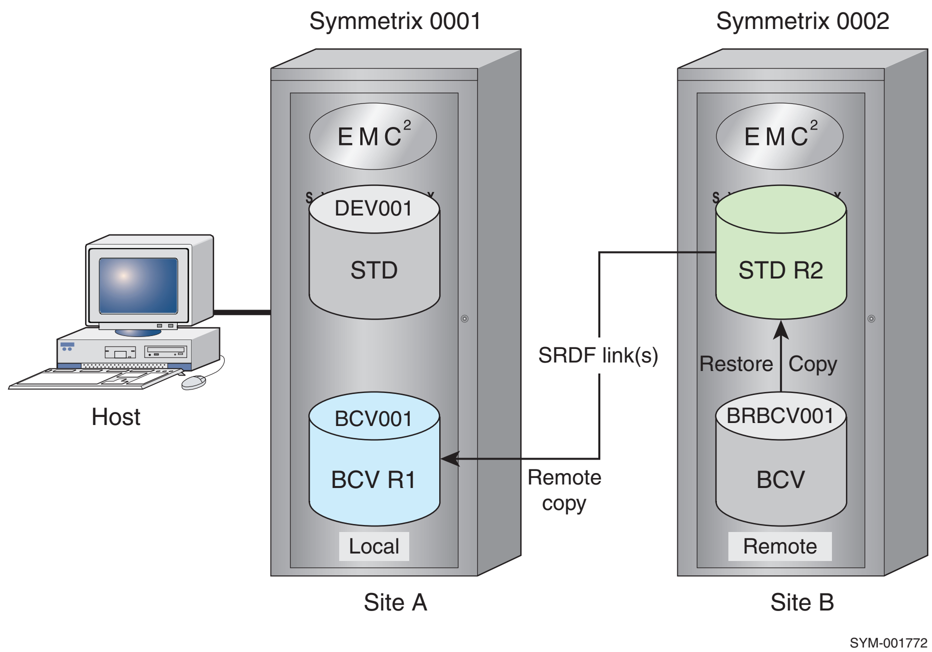

Figure 73 Remote copy with restore

It is recommended that whenever possible, this type of restore operation be performed at the remote site in two command steps (restore the BCV pair first, and then restore the R1 from the R2):

```
symmir -g DgName restore -bcv -rdf -full
symrdf -g DgName restore -bcv
```

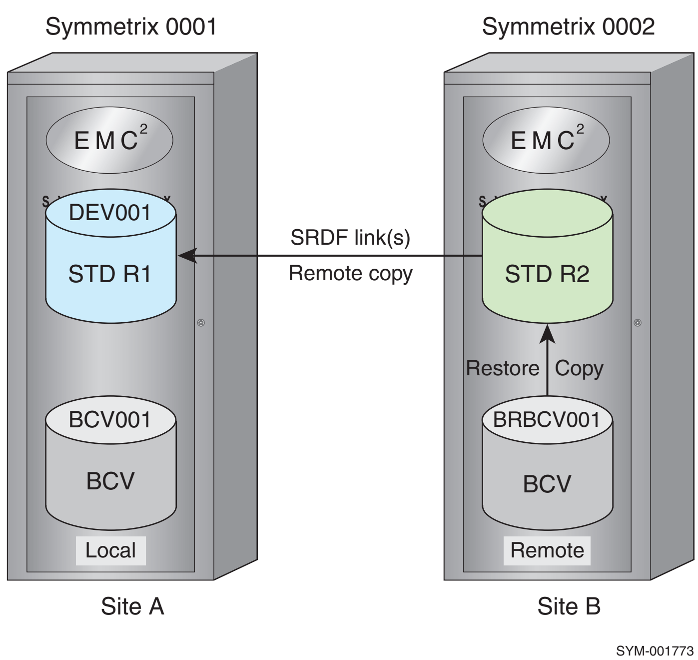

Figure 74 Restoring to an R1 device ready on the link

Incrementally restoring BCV pairs

The incremental restore process (Figure 75 on page 181) accomplishes the same thing as the restore process with a major time-saving exception: the BCV (BCV001) copies to the standard device (DEV001) only the new data that was updated on the BCV device while the BCV pair was split. Any changed tracks on the standard device are also overwritten by the data on the corresponding tracks on the BCV device. This maximizes the efficiency of the synchronization process.

This process is useful if the results from running a new application on the BCV device were desirable, and the user wants to port the data and the new application to the standard device.

The following forms enable you to target devices in a device group, composite group, or device file:

```
symmir -g DgName restore
symmir -cg CgName restore
symmir -f[ile] FileName restore
```

Note: For information on creating a device file, refer to "Device file" on page 216.

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

```
symmir -g prod restore
```

Note: It might be desirable for your site to set external device locks on all standard and BCV devices you are about to restore; refer to "Device external locks" on page 117.

To initiate an incremental restore on a BCV pair, DEV001, in the prod device group, enter:

```
symmir -g prod restore DEV001
```

To initiate an incremental restore on more than one (list) BCV pair in the prod device group, enter:

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

Figure 75 illustrates an incremental restore of a BCV pair.

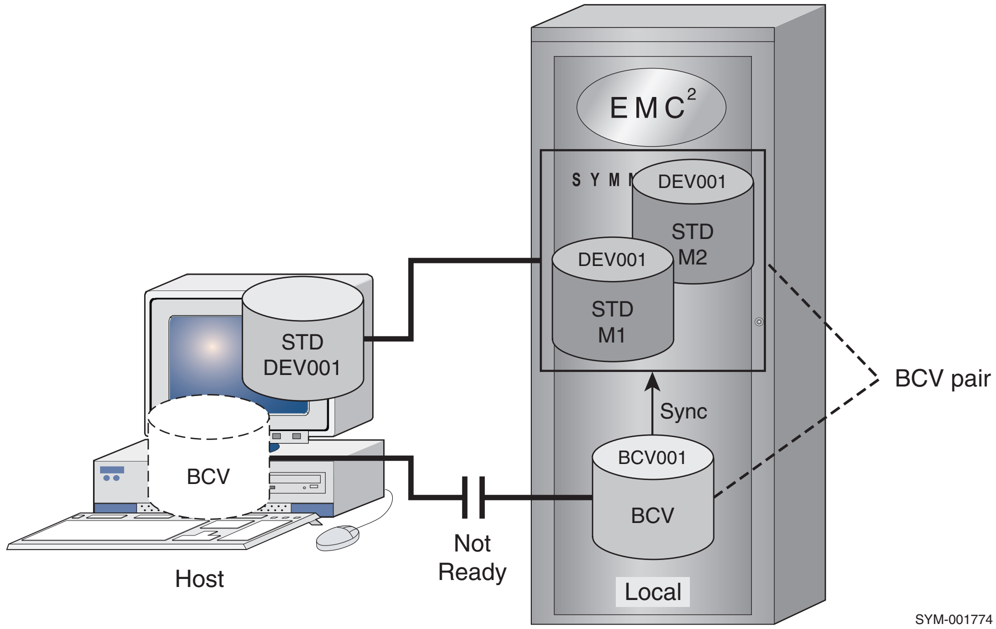

Figure 75 Incremental restore the STD

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

- Command validity is checked. For example, the command is rejected if the BCV device and the standard device were not previously paired.
- The BCV device is set as Not Ready to the host.
- The BCV device is assigned as the next available mirror of the standard device.
- The tracks are copied from the BCV device to the standard device. Any new data written to the BCV device while the BCV pair was split is written to the standard device. Any new data written to the standard device while the BCV pair was split is overwritten by the data on the corresponding track on the BCV device.

The BCV pair is synchronized when the standard device and the BCV device contain identical data.

Note: The BCV device is not available for host use while it is assigned as a BCV mirror on a standard device. However, any new data written to the standard device is copied to the BCV device while the BCV pair exists.

Incrementally restoring multiple BCV pairs

You can incrementally establish or restore up to 16 BCV pairs (8 pairs when using emulation) associated with a single standard device. Using the environment variable SYMCLI_MAX_BCV_PAIRS, the maximum amount of pairs can be adjusted from 1 to 16 BCV devices. If a series of `split/increment establish` commands were invoked over time (refer to "Incrementally establishing multiple BCV pairs" on page 148), a multi-BCV environment becomes established that retains progressive historical images of the specified standard.

With the incremental restore command, you can specify any one of these older BCVs to incrementally restore the standard back to a specific historical copy. Figure 76 shows the standard being restored by the `BCV002` copy, which was split at 4 a.m.

Before you restore, you may need to query the Symmetrix array to see which BCVs can be incrementally restored:

```
symmir -g DgName query -multi
```

CAUTION

Before invoking the restore command, be sure this is the data copy you want your standard restored to since this BCV is an older version of what is current.

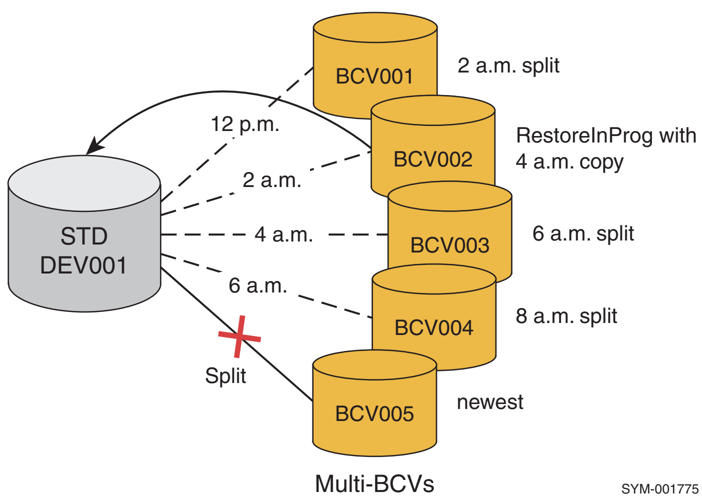

Figure 76 Restoring a BCV in a multi-BCV environment

For example, to restore the current standard with old data from `BCV002`, enter:

```
symmir -g DgName split DEV001
symmir -g DgName restore DEV001 BCV ld BCV002
```

Incrementally restoring SRDF-connected BCV pairs

You can also specify an incremental `restore` action to a remote Symmetrix site using the RDF flag (`-rdf` option), which incrementally restores the remote BCV pairs.

CAUTION

Be sure you want the standard R1 device to be updated with the restored copy before issuing the `-remote` option along with this restore operation.

For example, Figure 77 on page 184 illustrates an incremental restore operation in the remote Symmetrix array at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf restore
```

In this case, the flag indicates that the BCV device being restored is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local R1 standard device.

In addition to restoring the specified BCV pair, you can further specify the remote (`-remote`) option, which will propagate the restored copy from the BCV pair across the SRDF link to the R1 standard.

Note: Since Enginuity Version 5568, the `-remote` option is no longer supported for restoring from an R2 standard to an R1 BCV. Instead you should use the two step method illustrated in Figure 78 on page 185.

Note: The `-remote` option (illustrated in Figure 77 on page 184) is required when restoring to an R1 device that is ready on the link.

It is recommended that this type of restore operation be performed in two command steps (restore the BCV pair first, then restore the R1 from the R2):

```
symmir -g DgName restore -rdf
symmrdf -g DgName restore
```

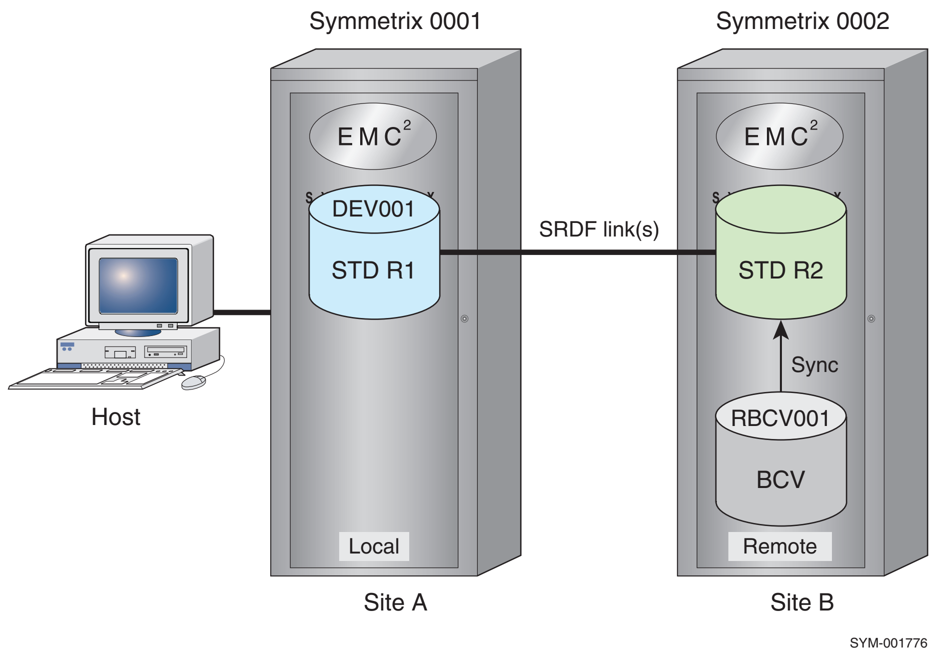

Figure 77 **Incrementally restoring the remote STD that mirrors local STD**

Figure 78 illustrates an incremental restore operation in the remote Symmetrix array at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv restore
```

In this case, the flags indicate that the BCV device being restored is an SRDF-connected BCV device, which will be established with the remote standard mirror of the local R1 BCV device.

In addition to restoring the specified BCV pair, you can further specify the remote (`-remote`) option, which will propagate the restored copy from the BCV pair across the SRDF link to the R1 BCV.

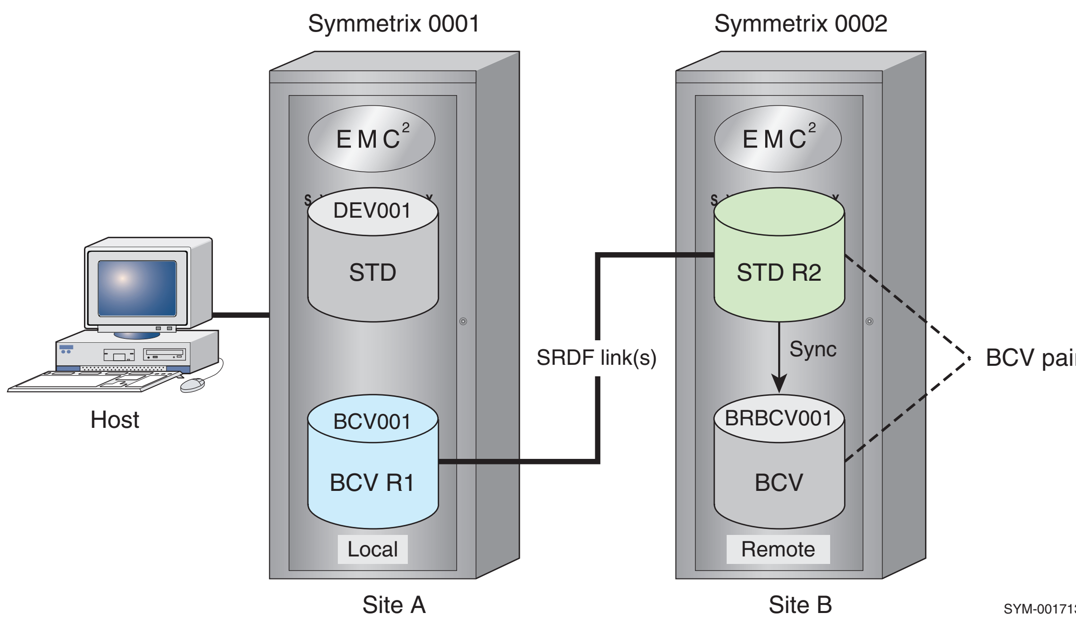

Figure 78 **Incrementally restoring the remote STD that mirrors a local BCV**

Incrementally restoring second-level remote BCV pairs

You can specify a incremental `restore` action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which incrementally restores second-level remote BCV pairs. Figure 79 illustrates a incremental restore operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv restore
```

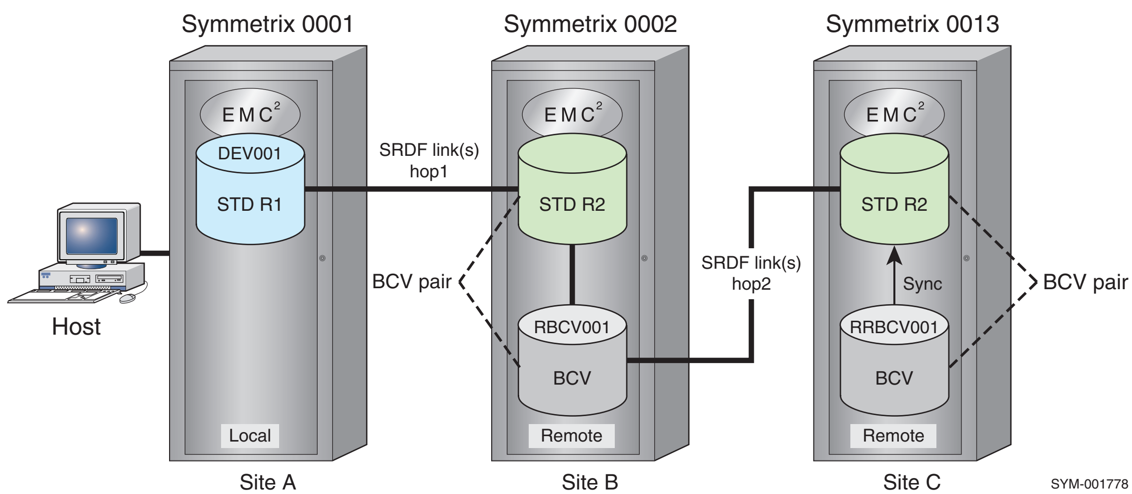

Figure 79 Incrementally restoring the remote hop 2 BCV pair

To initiate a restore on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv restore RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being restored is a second HOP SRDF-connected BCV device, which will be established with the remote standard mirror of the remote BCV device.

Incrementally restoring hop 2 BCV pairs in a Cascaded SRDF configuration

You can specify a incremental `restore` action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 80 illustrates an incremental restore operation in the remote Symmetrix array at the tertiary site (C) when the hop 2 flag (`-hop2` option) is specified with the following command:

```
symmir -g prod -hop2 restore
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being established with the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

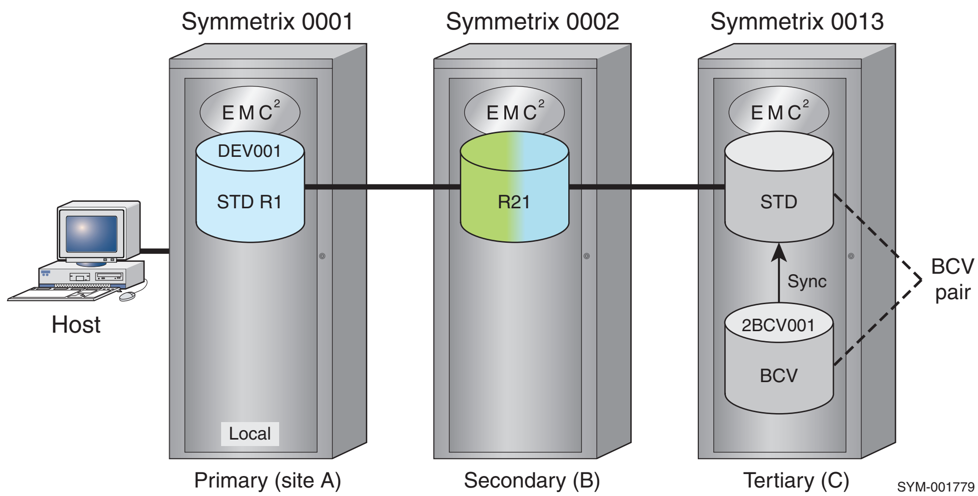

Figure 80 Incrementally restoring the hop2 standard in a Cascaded SRDF configuration

Protecting BCV data during full or incremental restores

Once you initiate a restore from the BCV to the standard device, data from the BCV is immediately available to a host accessing the standard device. During the time the pair are joined, writes are sent to the standard and the BCV, while reads are satisfied by the data on the BCV if the data on the standard has not yet been completely updated from the BCV. However, this process can alter the BCV data during the restore operation. If you want to retain the original BCV data, use the protected restore feature.

The protected restore feature allows the contents of a BCV to remain unchanged during and after a restore operation, even while the BCV and the standard are joined. Subsequently, any writes to the BCV pair are not propagated to the BCV while the standard and the BCV are joined in a RestInProg or Restored state. This protection offers the same advantage as a reverse split, but without the need for a mirrored BCV.

You can restore data from a BCV to a standard device without altering the contents of the BCV, by using the protect (`-protect`) option. It write-disables the BCV mirror(s) during and particularly after the restore operation.

To initiate a protected *full* restore (for example) on the STD mirrors (`DEV001`) in the `Prod` group, enter:

```
symmir -g Prod -full restore -protect DEV001
```

To initiate a protected *incremental* restore (for example) on the STD mirrors (`DEV001`) in the `Prod` group, enter:

```
symmir -g Prod restore -protect DEV001
```

Note: If you ever need to split a device again that was protected restored, you must use the `-protect` option on the split command:

```
symmir -g Prod split -protect DEV001
```

To view device information for the protected restore operation, enter:

```
symmir -g Prod query -protect
```

Note: The standard invalid track count displayed in the query operation does not reflect any new writes while the device is in the RestInProg state. When the device state changes to Restored, the invalid track count displays as zero.

Cancelling BCV pairs

The `symmir cancel` command allows you to cancel a BCV pair relationship on a device by device basis, or for all the devices in a device group or composite group.

When operating in native TimeFinder, cancelling a BCV pair cancels the existing relationship between the specified standard and BCV device(s). Once the relationship is cancelled, the corresponding BCV devices go into the SplitNoInc state, and the BCV pair can no longer be incrementally established or restored.

When operating in emulation mode, cancelling a BCV pair terminates the relationship between the specified standard and BCV device(s). Once the relationship is terminated, the corresponding BCV devices go into the Never Established state, and the BCV pair can no longer be incrementally established or restored.

When cancelling a muli-BCV relationship, only the primary BCV is cancelled. For information on cancelling a multi-BCV relationship, refer to "Canceling a multi-BCV relationship" on page 136.

The following forms enable you to target devices in a device group, composite group, or device file:

```
symmir -g DgName cancel
symmir -cg CgName cancel
symmir -f[ile] FileName cancel
```

To cancel the BCV relationship for all the devices in the `Prod` group, enter:

`symmir -g Prod cancel`

To cancel a specific standard/BCV pair relationship in the `Prod` group, enter:

`symmir -g Prod cancel DEV001 BCV dev 009C`

To cancel the relationship of SRDF-connected BCV pairs in the `Prod` device group, enter any of the following:

`symmir -g prod cancel -rdf`

Cancels the relationship between the remote mirror device(s) and the remote BCV device(s).

`symmir -g prod cancel -rdf -bcv`

Cancels the relationship between the SRDF-connected BCV pair remotely mirroring the local BCV device.

`symmir -g prod cancel -rrbcv`

Cancels the relationship between the remote mirror of the remotely attached BCV device (RBCV) and the remotely attached remote BCV (RRBCV).

`symmir -g prod cancel -hop2`

Cancels the relationship between the remote mirror and the BCV (2BCV) two hops away in a Cascaded SRDF configuration.

Querying BCV pairs

You can perform a query to determine the state of a BCV pair or all BCV pairs in a device group, composite group, or device file. The query is sent via the gatekeeper device to the Symmetrix array, returning with information about the state of the BCV pair(s).

The following forms enable you to target devices in a device group, composite group, or device file:

```
symmir -g DgName query
symmir -cg CgName query
symmir -f[ile] FileName query
```

Note: For information on creating a device file, refer to "Device file" on page 216.

For example, to query the state of the BCV pairs in the `prod` device group, enter:

```
symmir -g prod query
```

To query the state of SRDF-connected BCV pairs in the `prod` device group, enter any of the following:

```
symmir -g prod query -rdf
symmir -g prod query -rdf -bcv
symmir -g prod query -rrbcv
symmir -g prod query -hop2
```

You can also obtain results using the `-offline` option, which looks at your configuration based on the Symmetrix host database.

The results of the query include the following information for each member of a BCV pair in a device group:

- Logical device name
- Symmetrix device name
- Number of invalid tracks
- BCV pair state

To query the state of a split action on multi-BCVs or concurrent BCVs in a group `prod`, enter:

```
symmir -g prod query -multi
```

To query the state of any background split action on multi-BCVs or concurrent BCVs in a group `prod`, enter:

```
symmir -g prod query -multi -bg
```

To query the percent initiated on restore, establish, and split operations, enter:

```
symmir -g prod query -bg -percent
```

Verifying BCV pair states

You can use the `symmir verify` command to verify whether one or all BCV pair(s) in a device group, composite group, or device file are in a particular state. The command can be used in scripts to guarantee that the BCV device pair(s) are in a Synchronized, Restored, or Split state prior to executing subsequent SYMCLI commands. If you do not specify any qualifiers with the `symmir verify`, the default is to check for the Synchronized or Restored states.

The following forms enable you to target devices in a device group, composite group, or device file:

```
symmir -g DgName verify
symmir -cg CgName verify
symmir -f[ile] FileName verify
```

Note: For information on creating a device file, refer to "Device file" on page 216.

The following options qualify the `symmir verify` command. If you need to verify a concurrent BCV pair, include `-concurrent` with the option (for example, `-synched -concurrent`):

- The `-synched` option verifies the Synchronized state.
- The `-syncinprog` option verifies the SyncInProg state.
- The `-split` option verifies the Split state. With an instant split, the system verifies the Split state immediately even though the background split is still in progress. To verify completion of a background split after an instant split, use the `-split -bg` option. Until the background split is complete, you cannot perform BCV control operations. You can use the `-split -bg` option to verify that the instant split is 100 percent complete in the background. For example:

  ```
  symmir verify -g ProdBgrp -split -bg DEV001 bcv ld BCV002 -i 30
  ```

- The `-restored` option verifies the Restored state. You can use the `-restored -protect` option to verify the Protected Restored state. In a concurrent BCV setup, you can use `-restored -concurrent` successfully only if the first BCV has already restored the standard and you are restoring now with the second BCV.
- The `-restinprog` option verifies the RestInProg state.
- The `-bcv_mirrors` option verifies that the mirrors of locally mirrored BCV devices are in the specified state. If you do not specify a state with this option, the default is to verify a Synchronized state.

For a multi-BCV or concurrent BCV device group, specifying the BCV on the command line ensures that the verify operation checks the status of the BCV. Otherwise, the verify operation checks the status of the standard device, which may no longer be established with the BCV that you want to verify. For example, the following command returns the status of standard device DEV002 with its last paired BCV:

```
symmir -g ProdBgrp verify DEV002
```

But the following command returns the status of a specific BCV pair (`DEV002` with `BCV001`):

```
symmir -g ProdBgrp verify DEV002 BCV ld BCV001
```

The following command checks status every 30 seconds until all BCV pairs in the device group (ProdBgrp) or composite group (MyConGrp) are in the Synchronized or Restored state (the default when no state is specified on the command line):

```
symmir -g ProdBgrp -i 30 verify
symmir -cg MyConGrp -i 30 verify
```

Possible outputs at 30-second intervals can be that none, not all, or all devices are synchronized or restored. The time to reach the Synchronized or Restored state varies with the number of devices being established or restored and the amount of data being copied.

The verify action returns a value of zero (code symbol CLI_C_SUCCESS) if the verify criteria are met, or one of the unique codes in Table 14 and Table 15 if the verify criteria are not met:

Table 14 Using options to verify a BCV mirror state

Options used with Verify	Code number	Code symbol
-bcv_mirrors	4	CLI_C_NOT_ALL_SYNCHRONIZED
-bcv_mirrors	5	CLI_C_NONE_SYNCHRONIZED
-bcv_mirrors -ready	62	CLI_C_NOT_ALL_READY
-bcv_mirrors -ready	63	CLI_C_NONE_READY
-bcv_mirrors -syncinprog	27	CLI_C_NOT_ALL_SYNCINPROG
-bcv_mirrors -syncinprog	28	CLI_C_NONE_SYNCINPROG
-bcv_mirrors -restinprog	29	CLI_C_NOT_ALL_RESTINPROG
-bcv_mirrors -restinprog	30	CLI_C_NONE_RESTINPROG

Table 15 lists the options to verify a BCV pair state.

Table 15 Using options to verify a BCV pair state

Options used with Verify	Code number	Code symbol
-synched	10	CLI_C_NOT_ALL_SYNCHED
-synched	11	CLI_C_NONE_SYNCHED
-restored	12	CLI_C_NOT_ALL_RESTORED
-restored	13	CLI_C_NONE_RESTORED
-split or -split -bg	25	CLI_C_NOT_ALL_SPLIT
-split or -split -bg	26	CLI_C_NONE_SPLIT
-syncinprog	27	CLI_C_NOT_ALL_SYNCINPROG
-syncinprog	28	CLI_C_NONE_SYNCINPROG
-restinprog	29	CLI_C_NOT_ALL_RESTINPROG
-restinprog	30	CLI_C_NONE_RESTINPROG

For example, to verify the state of the BCV pairs in the `prod` device group, enter:

```
symmir -g prod verify
```

To verify the state of SRDF-connected BCV pairs in the `prod` device group, enter any of the following:

```
symmir -g prod -rdf verify
symmir -g prod -rdf -bcv verify
symmir -g prod -rrbcv verify
symmir -g prod -hop2 verify
```

Note: Note that if you use the `-force` option with the above commands, the returned status may differ. Without the `-force` option, all TimeFinder devices (established or not) are examined with this command. With the `-force` option, only the TimeFinder devices that have a pairing relationship (that were established) are examined.

Preventing host access

At the end of a split operation, the BCV becomes accessible to the host. To make the BCV inaccessible to the host after the split, use the `split -not_ready` option. Similarly, using the `-not_ready` option with a restore action makes the standard inaccessible upon initiating the restore operation. There are several situations where using this option may be useful:

- When a BCV is established with a standard, all information (including volume manager specific information such as a Windows 2000 device signatures) is copied to the BCV. While the standard and BCV are synchronized, this is not a problem because the BCV is not ready (not visible to the host). However, when the BCV is split, if the host accessing the standard can also access the now ready (host-visible) BCV, the Windows Disk Administrator incurs a problem. When it detects a duplicate signature, Windows 2000 arbitrarily places one of the drives offline. To avoid this problem, you can make the BCV device temporarily inaccessible while in the split condition by using a `symmir split -not_ready` command. Subsequently, after relabelling the BCV with a new signature, you can use the `symld ready` command to make the BCV accessible again.
- A similar situation arises when a restore operation overwrites the contents of the standard volume (including the signature) with the contents of the BCV. Using the restore `-not_ready` action allows you to relabel the standard before making it accessible to the host. The `-not_ready` option is also useful when you want to wait until the restore operation completes before using the standard.
- When TimeFinder control operations are performed on a Symmetrix array by more than one process, regardless of whether they are executing on one or more hosts, there is a danger that one application will accidentally usurp a BCV that contains data belonging to a different application. Using the `split -not_ready` command can make it easier to prevent accidental misuse.

The `symlabel` command works on Symmetrix Fixed Block Architecture (FBA) devices on Windows 2000 to avoid the label conflict problem described above. To examine the emulation mode of a Symmetrix device, use the `symdev show` command. The following sequence outlines how to establish and split a BCV pair (DEV001/BCV001) and suspend read/write operations to the BCV until you have relabeled the BCV device:

1. Define device signatures for the standard and BCV devices. The signature that you specify must be an 8-character hexadecimal value (for example, 12ABCDEF). The `-type` parameter specifies that the operation is on a Windows 2000 system. If you omit the label option as in the BCV001 signature definition, SYMCLI reads the signature off the disk and stores it in the SYMAPI database:

```
symlabel -g ProdBgrp define -type WNT DEV001 label 12ABCDEF
symlabel -g ProdBgrp define -type WNT BCV001 -bcv
```

2. Establishing the BCV pair causes the BCV signature to become identical to the standard signature:

```
symmir -g ProdBgrp -full establish DEV001 bcv ld BCV001
```

3. After synchronization, split the pair so that the BCV device is in the Not Ready state after the split:

```
symmir -g ProdBgrp split -not_ready
```

4. Relabelling the BCV device causes the BCV signature to revert to its original signature that was stored in the SYMAPI database:

   ```
   symld -g ProdBgrp relabel -bcv
   ```

5. Change the status of the BCV device to Ready:

   ```
   symld -g ProdBgrp ready -bcv
   ```

Using composite groups to manage BCV pairs across Symmetrix arrays

A composite group is a user-defined group of devices that can span multiple Symmetrix arrays. This feature provides greater flexibility than a device group, which can define devices only on a single Symmetrix array. You can control specific BCV pairs within the composite group instead of having to operate on the entire group as in previous versions.

Figure 81 illustrates a production host that is locally connected to two Symmetrix arrays (A and B). A composite group is defined on the production host and includes BCV pairs from each Symmetrix array. Another locally connected host allows you to access the BCVs once the BCV pairs are split.

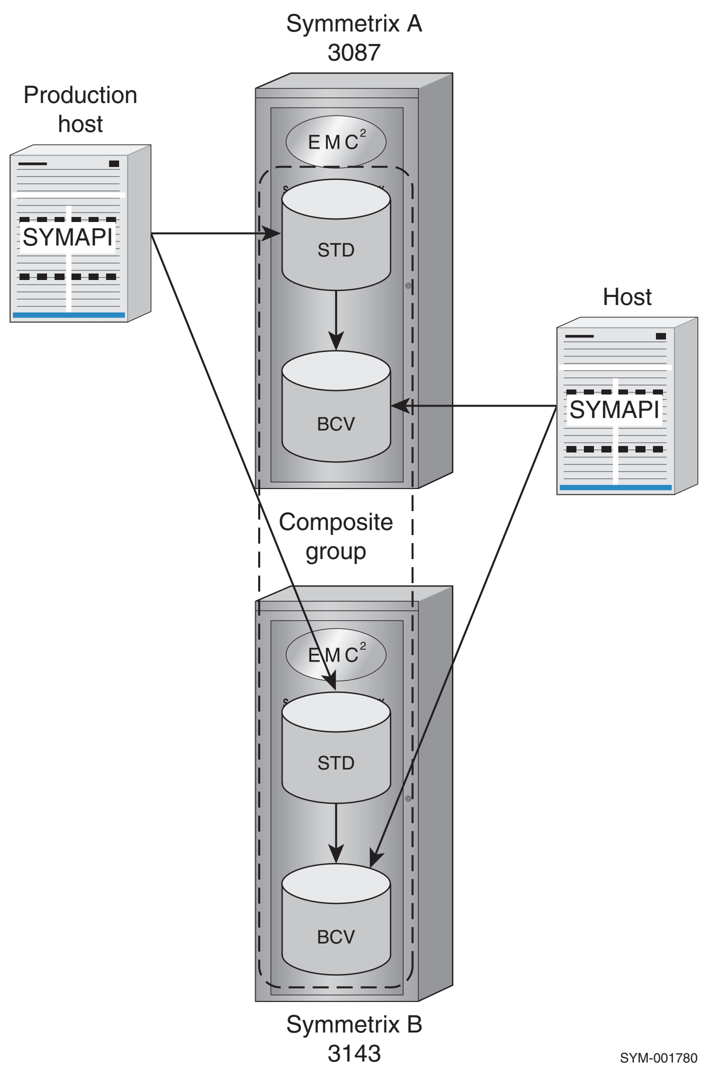

Figure 81 Using a composite group when a set of devices spans two Symmetrix arrays

Although TimeFinder control operations on BCV pairs might normally be performed from the production host (as shown in Figure 81) because the composite group is defined there in its SYMAPI database, there are methods that would allow you to initiate copy sessions from another locally connected host. One way is to copy the composite group definition to another host. A more efficient method is to enable Group Name Services (GNS), which automatically propagates the composite group definition to the Symmetrix arrays and other locally attached hosts that are running the GNS daemon. For more information, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

Note: Prior to Solutions Enabler Version 6.0, composite group support did not allow you to copy from a single source standard device to more than one BCV (that is, not to concurrent BCVs). Consequently, you had to add an equal number of standard devices and BCV devices to the composite group. Solutions Enabler Version 6.0 lifts this restriction by allowing you to create each BCV pair explicitly, including concurrent BCVs.

If you do not create each BCV pair explicitly, certain options such as `-opt`, `opt_rag`, and `-exact` allow you to control how multiple devices in a composite group are paired. Otherwise, a device-pairing algorithm checks if there were any previous pair assignments among the devices and, if not, pairs standards and BCVs of equal sizes.

Used only with a full establish operation, the optimize option that you choose depends on whether you are establishing local BCV pairs or remote BCV pairs. The `-opt` option is for local. It optimizes pairings across the local Symmetrix array without regard for whether the devices belong to different RDF (RA) groups. The `-opt_rag` option is for remote and requires the `-rdf` option. It uses optimization rules to create remote BCV pairs from devices within the same RDF (RA) group on a Symmetrix array.

The following steps outline the setup required for controlling a set of BCV pairs that spans two Symmetrix arrays as shown in Figure 81 on page 196:

1. From the production host, create a Regular type composite group (for example, `MyGrp`):

   ```
   symcg create MyGrp -type regular
   ```

2. Add to the composite group those standard devices on Symmetrix A (3087) and Symmetrix B (3143) that are the source devices:

   ```
   symcg -cg MyGrp -sid 3087 add dev 0076
   symcg -cg MyGrp -sid 3143 add dev 0091
   ```

3. Add a BCV device from each Symmetrix array to the composite group:

   ```
   symbcv -cg MyGrp -sid 3087 associate dev 0051
   symbcv -cg MyGrp -sid 3143 associate dev 004F
   ```

4. Create the BCV pairs and initiate full copying from the standards to the BCVs:

   ```
   symmir -cg MyGrp establish -full
   ```

5. When the BCV pairs are fully synchronized, you can split all BCV pairs in the composite group in order to access the BCVs:

   ```
   symmir -cg MyGrp split
   ```

You can control specific BCV pairs within the composite group instead of having to control the group as a whole. For example, to establish only the `DEV001/ BCV001` pair from all devices in the composite group `MyGrp`:

```
symmir -cg MyGrp establish DEV001 bcv ld BCV001
```

Preferred attachment of BCVs (optional operations)

For advanced users, the preferred pair *attachment* (`attach` action) is an optional step in the management of BCV pairs that eliminates the need to specify a device for each subsequent full establish and full restore sequence in a script (for all Enginuity versions). It also applies to incremental establish and restore operations. It marks the specified BCV device as the preferred BCV to pair with the standard device.

After configuration and initialization of a Symmetrix array, BCV devices contain no data. The BCV devices, like the standard devices, have unique host addresses and are online and ready to the host(s) to which they are connected.

It is at this point, before any full establish or full restore operations are requested, you can validate your pairings as a preferred attachment before starting any data copy operations. The lists of individual standard devices and BCV devices can be examined, validated, and all devices sorted according to storage size, and subsequently, assigned as the preferred match (considering disk size) for attachment into BCV pairs.

Note: A full establish action with the optimize (`-opt)` or exact (`-exact`) option overrides the attach pairing scheme.

The following commands enable you to target devices in a device group, composite group, or a device file:

```
symmir -g DgName attach
symmir -cg CgName attach
```

To initiate a preferred attachment on a BCV pair (`DEV001`) in the `prod` group, enter:

```
symmir -g prod attach DEV001 BCV ld BCV001
```

To initiate a preferred attachment on more than one BCV pair (list) in the `prod` group, enter:

```
symmir -g prod attach  DEV001 BCV ld BCV001
                       DEV002 BCV ld BCV002
                       DEV002 BCV ld BCV003
```

Note: The attach and detach preferred relationship are only known to the SYMAPI database on which you are operating.

Figure 82 illustrates a preferred attachment of a BCV pair.

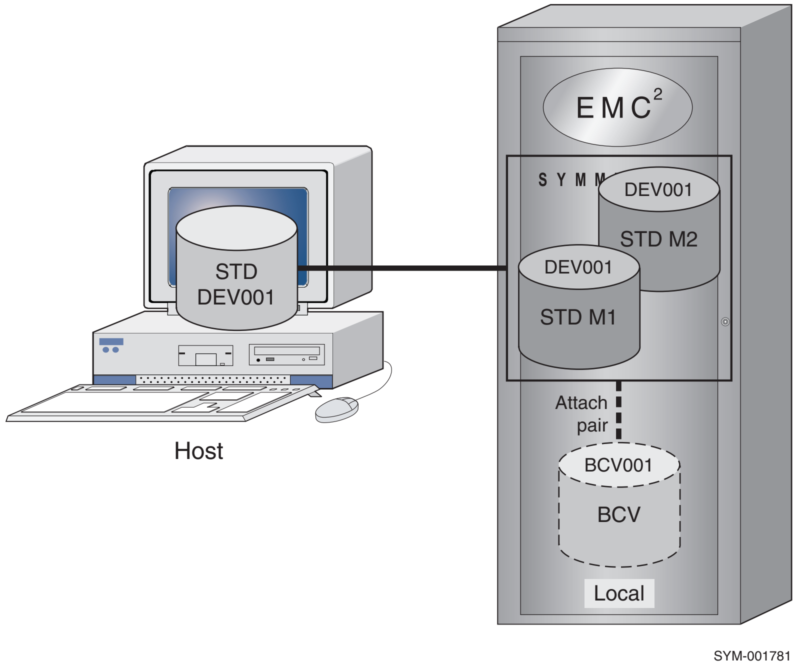

Figure 82 **Preferred attachment for BCV pairing**

The attach action checks command validity. For example, the Symmetrix array makes sure that both the standard device and the BCV device are the same size, the device specified as the BCV has the BCV attribute, the standard device does not already have a BCV device assigned to it, and so on.

If the standard device is a meta head device, then the BCV must also share the same meta device properties. All meta members are implicitly established along with the meta head device.

From this point forward, when you invoke the full establish or full restore control action with a BCV control operation, you will not need to specify the device names.

Detaching BCV preferences from devices

The `detach` action allows you to remove the preferred matched-pair association from the devices that was initially defined with the `attach` action.

The following forms enable you to target devices in a device group, composite group, or device file:

```
symmir -g DgName attach
symmir -cg CgName attach
```

For example, to detach the existing preferred attachment of various BCVs from their standard devices in the `prod` group, enter:

```
symmir -g prod detach
```

To detach the attached BCV preference on standard device (`DEV001`) in the `prod` group, enter:

```
symmir -g prod detach DEV001
```

Figure 83 illustrates a detachment of a BCV pair preference.

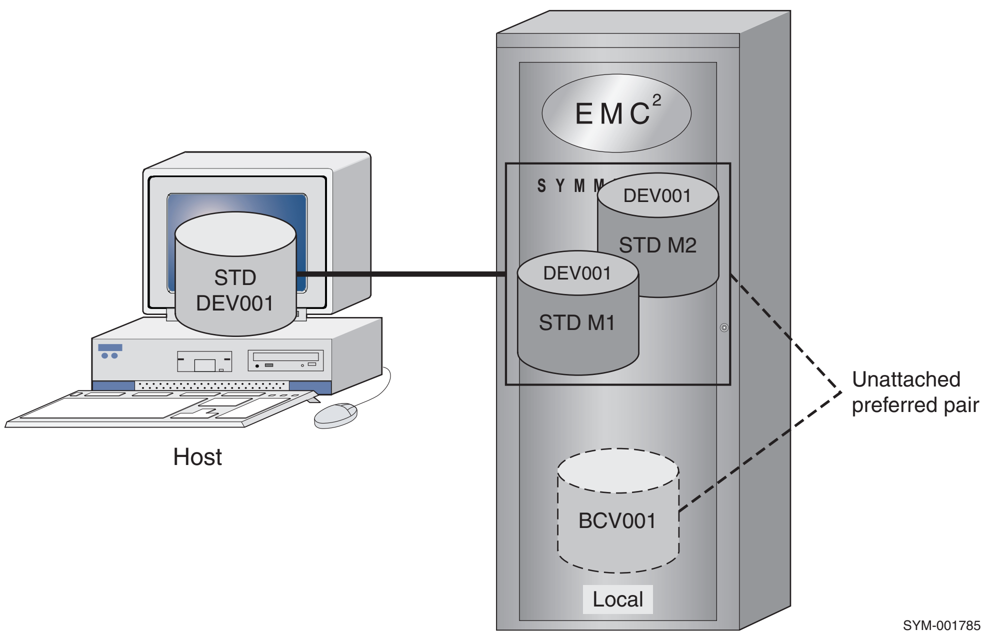

Figure 83 Detaching a BCV preference

Attaching remote devices as preferred pairs

For advanced usage, you can also specify an `attach` action to a remote Symmetrix site using the RDF flag (`-rdf` option), which attaches the remote mirror device(s) to the remote BCV device(s) as preferred pair(s).

For example, Figure 84 illustrates a preferred attachment operation in the remote Symmetrix array at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf attach DEV001
             bcv ld RBCV001
```

In this case, the RDF flag indicates that the BCV device being attached is an SRDF-connected BCV pair, which will provide remote mirroring to the local standard device.

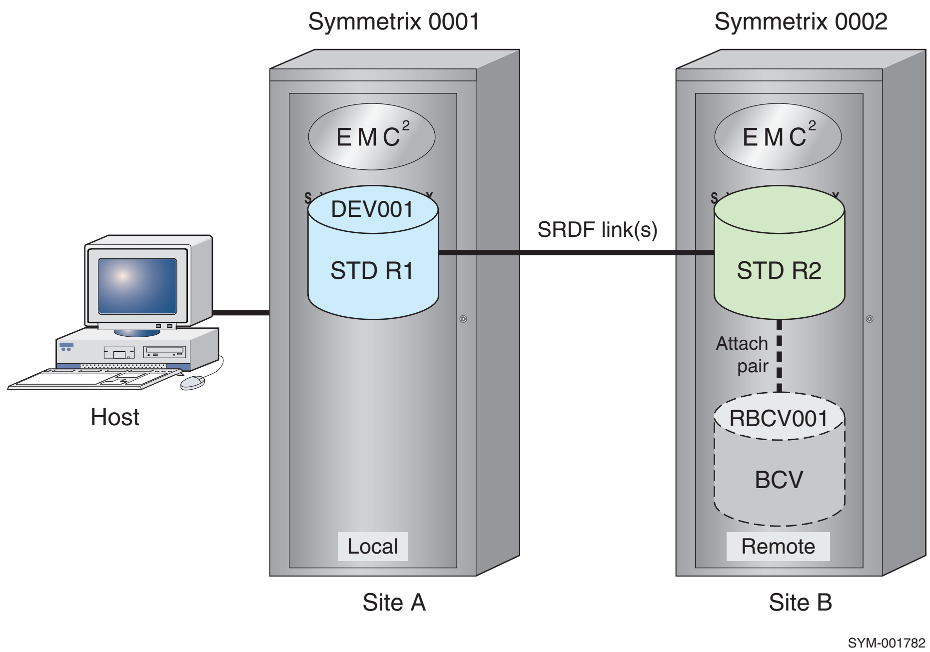

Figure 84 Attaching remote BCV to mirror local STD as a preferred pair

Figure 85 illustrates an attach operation in the remote Symmetrix array at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv attach BCV001 BCV ld BRBCV001
```

In this case, the flags indicate that the BCV pair being attached is an SRDF-connected BCV pair, which provides remote mirroring to the local BCV device.

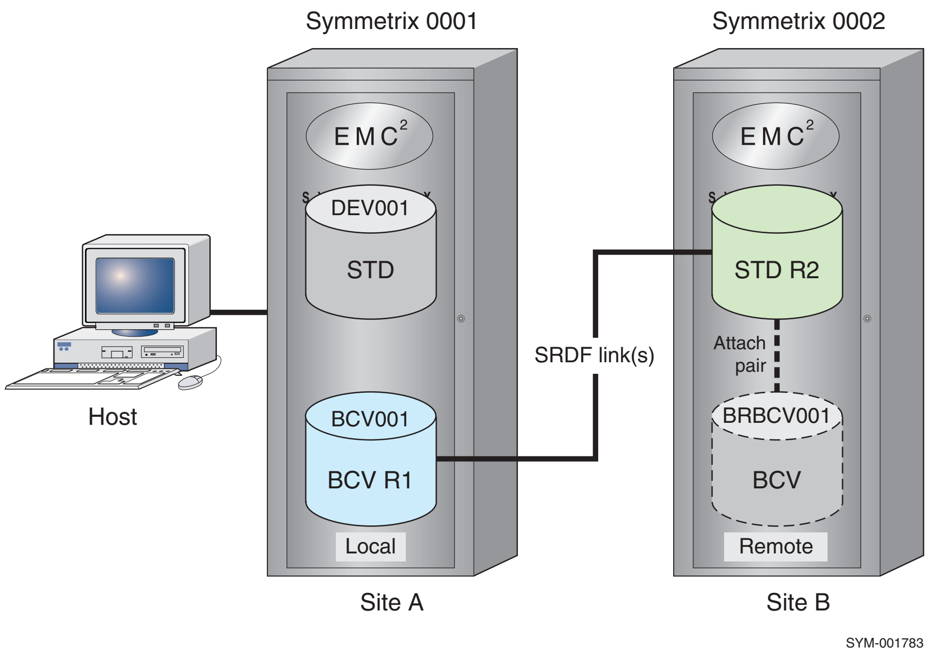

Figure 85 Attaching remote BCV to mirror local BCV as preferred pair

Detaching BCV preferences for remote devices

You can specify a `detach` preference action to a remote Symmetrix site using the RDF flag (`-rdf` option), which detaches the remote BCV(s) from the remote standard device(s) as preferred pair(s).

For example, Figure 86 illustrates a detach operation in the remote Symmetrix array at site B when the RDF flag is specified with the following command:

```
symmir -g prod -rdf detach DEV001
```

In this case, the flag indicates that the preferred BCV device being detached is an SRDF-connected BCV, which provides remote mirroring to the local standard device.

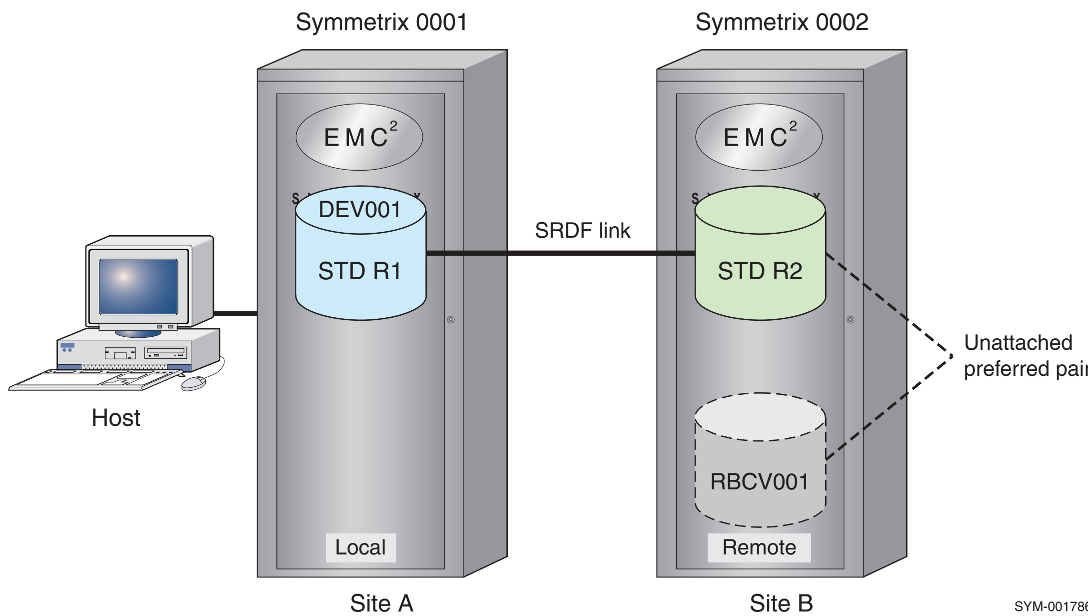

Figure 86 Detaching a BCV preference from remote STD (mirrored local STD)

Figure 87 illustrates a detach operation in the remote Symmetrix array at site B when the RDF and BCV flags (`-rdf` and `-bcv` option) are specified with the following command:

```
symmir -g prod -rdf -bcv detach BCV001 BCV ld BRBCV001
```

In this case, the flags indicate that the BCV pair being detached is an SRDF-connected BCV pair, which would provide remote mirroring to the local BCV device.

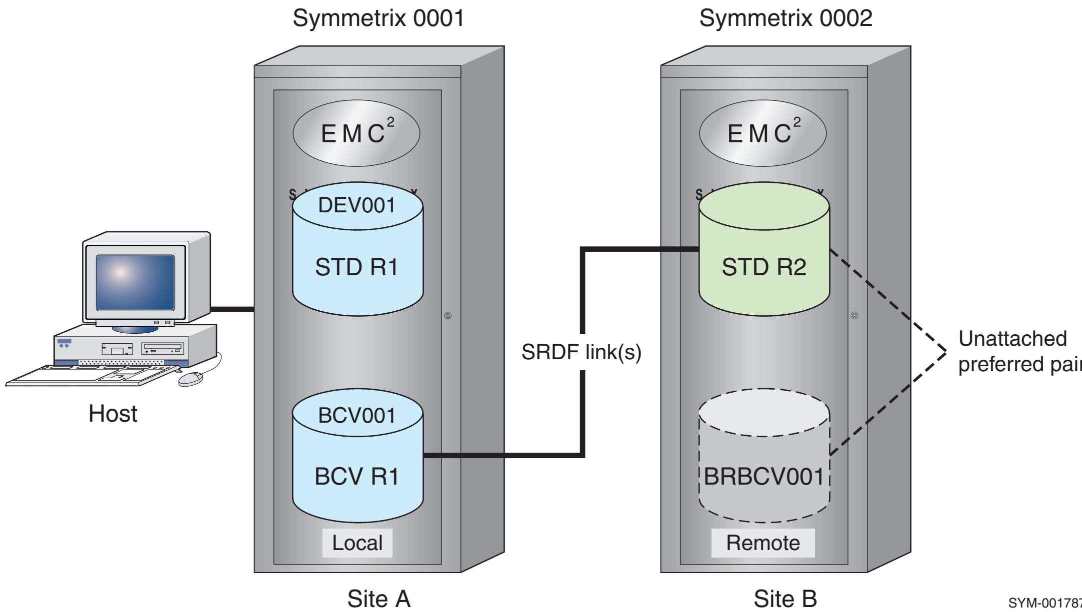

Figure 87 Detaching a BCV preference from remote STD (mirrored local BCV)

Attaching second-level remote devices as preferred pairs

You can specify an `attach` action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which attaches BCV preferences to second-level remote BCV pairs. Figure 88 illustrates a attach operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv attach
```

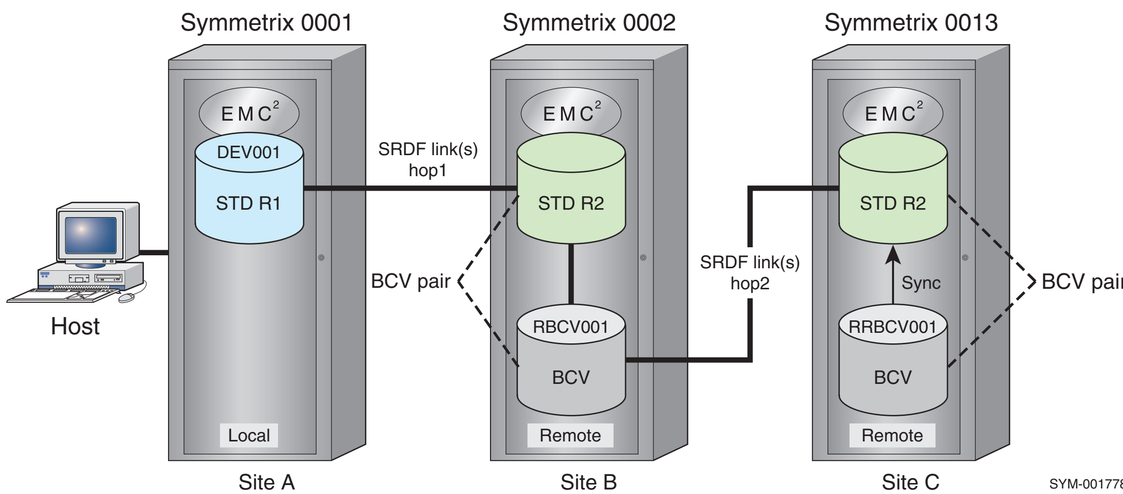

Figure 88 Attaching second level remote devices as preferred pairs

To initiate an attach on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv attach RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being attached is a second HOP SRDF-connected BCV device, which will be attached with the remote standard mirror of the remote BCV device.

Detaching BCV preferences from second-level remote devices

You can specify a `detach` action to a second remote Symmetrix site using the remotely attached remote BCV flag (`-rrbcv` option), which detaches BCV preferences from second-level remote BCV pairs. Figure 89 illustrates a detach operation in the remote Symmetrix array at site C when the remotely attached remote BCV flag (`-rrbcv` option) is specified with the following command:

```
symmir -g prod -rrbcv detach
```

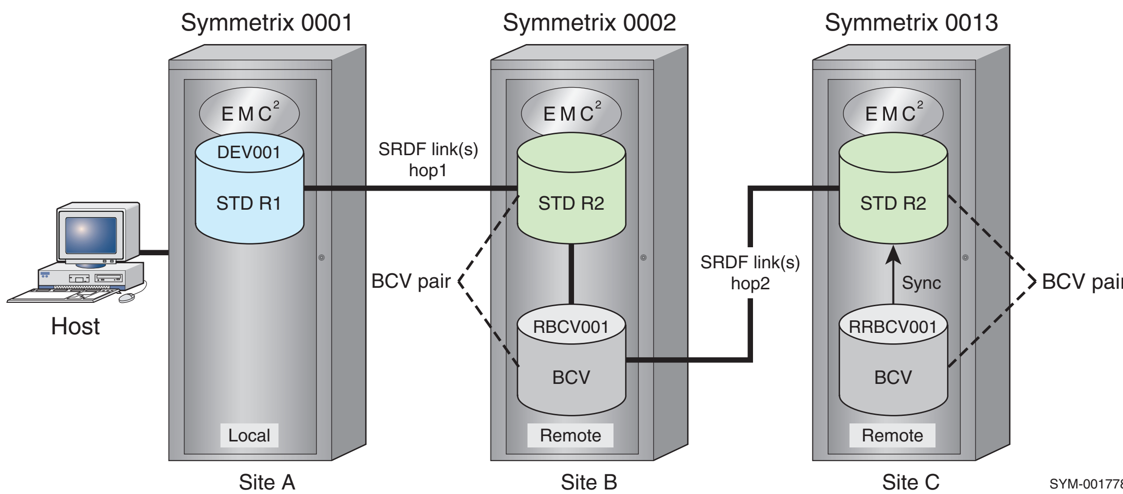

Figure 89 Detaching a BCV preference from a second level remote STD

To initiate a detach on one remote BCV pair, `RBCV001`, in the `prod` group, enter:

```
symmir -g prod -rrbcv detach RBCV001 BCV ld RRBCV001
```

In this case, the flag indicates that the BCV device being detached is a second HOP SRDF-connected BCV device, which will be detached from the remote standard mirror of the remote BCV device.

Attaching hop 2 devices as preferred pairs in a Cascaded SRDF configuration

For advanced usage, you can specify an `attach` action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 90 illustrates an attach operation in the remote Symmetrix array at the tertiary site (C) when the hop2 (`-hop2` option) is specified with the following command:

```
symmir -g prod -hop2 attach DEV001
             bcv ld 2BCV001
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being attached with the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

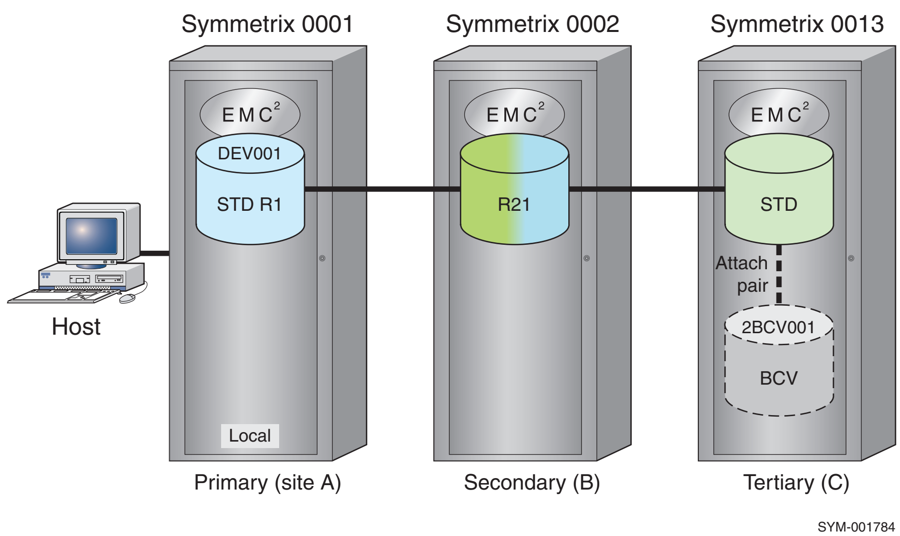

Figure 90 Attaching hop 2 devices as preferred pairs in a Cascaded SRDF configuration

Detaching BCV preferences from hop 2 devices in a Cascaded SRDF configuration

For advanced usage, you can also specify an `detach` preference action to a Symmetrix array located at the tertiary site of a Cascaded SRDF configuration.

For example, Figure 91 illustrates a detach operation in the remote Symmetrix array at the tertiary site (C) when the hop 2 (`-hop2` option) is specified with the following command:

```
symmir -g prod -hop2 detach DEV001 BCV ld 2BCV001
```

In this case, the flag indicates that the SRDF-connected BCV device (2BCV001) is being detached from the remote partner of the R21 device, which is the remote partner of the local RDF standard device.

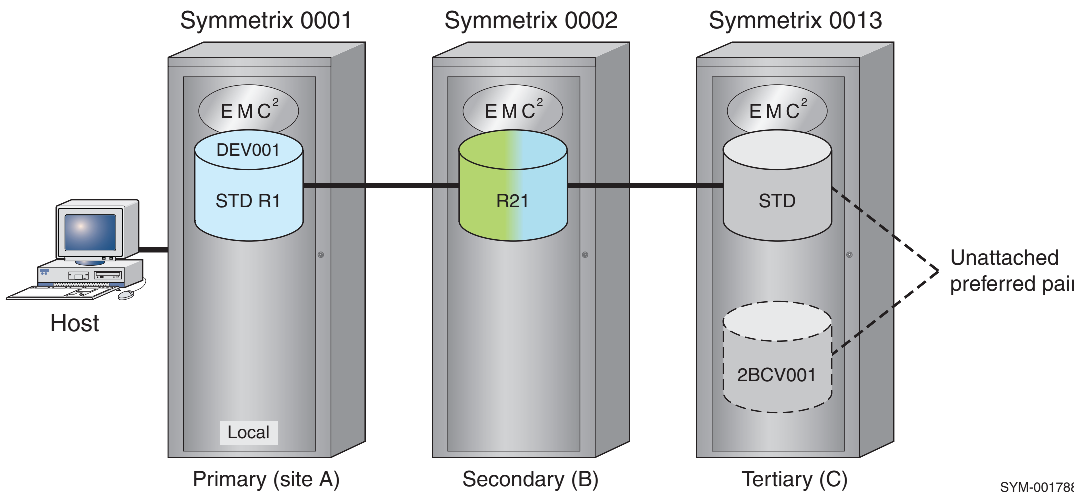

Figure 91 Detaching preferences from hop 2 devices in a Cascaded SRDF configuration

Script summary for typical TimeFinder operations

The following is an example script of a set of typical operations using SYMCLI commands to manage a BCV environment:

1. Create a device group:

   `symdg create ProdBgrp`

2. Add a standard device to a device group:

   `symld -g ProdBgrp add pd c0t2d4`

   Repeat for all devices, or use RANGE, etc.

3. Associate a BCV device with a device group:

   `symbcv -g ProdBgrp associate pd c4t2d4`

   Repeat for all BCVs.

4. Either establish the entire group:

   `symmir -g ProdBgrp -full establish -noprompt`

   Or establish explicitly:

   `symmir -g ProdBgrp -full establish DEV001 BCV ld BCV001 -noprompt`

   Repeat this command for all pairs.

Transfer a different BCV device to the standard device:

1. Identify the established BCV pair.
2. Split the pair:

   `symmir -g ProdBgrp split DEV001 -noprompt`

3. Select a new BCV to establish with the standard device:

   `symmir -g ProdBgrp -full establish DEV001 BCV ld BCV020 -noprompt`

Script example for multi-BCV environment

The following is an example of a script for a multi-BCV environment:

You are tasked with testing business applications with incoming database data from certain anticipated peak periods in the day. Three copies of the database may be needed.

To establish a multi-BCV environment, you must initially perform a full establish to each BCV device in the set.

1. For example, you plan to have `BCV001` through `BCV003` in the set to pair with `DEV001` that is the source of your test data:

   ```
   symmir -g MultigrpA -full establish DEV001 BCV ld BCV001

   symmir -g MultigrpA split DEV001 #split at 3:10pm

   symmir -g MultigrpA -full establish DEV001 BCV ld BCV002

   symmir -g MultigrpA split DEV001 #split at 3:20pm

   symmir -g MultigrpA -full establish DEV001 BCV ld BCV003
   ```

2. It is now 4:00 p.m. and `BCV003` is still currently established with `DEV001`. At this point, you are testing your business applications and want to reset your database back to the business activity that was current up till 3:10 p.m. To incrementally restore `DEV001` to the 3:10 p.m. business data:

   ```
   symmir -g MultigrpA split DEV001 #split at 4:00pm
   symmir -g MultigrpA restore DEV001 BCV ld BCV001
   ```

3. You are now working successfully with the 3:10 p.m. data and want to continue test operations with this data and remove the second split BCV that occurred at 3:20 p.m. as this data will not be needed.

   ```
   symmir -g MultigrpA cancel DEV001 BCV ld BCV002
   ```

4. You decide to call it a day, keeping the remaining two multi-BCVs and need to reestablish (incrementally establish) `BCV003` to the current standard data:

   ```
   symmir -g MultigrpA split DEV001 #split at 4:50pm
   symmir -g MultigrpA establish DEV001 BCV ld BCV003
   ```

BCV pair states

When you invoke BCV control commands on a single BCV device, or on a group of BCV pair(s) using the `symmir` command, the BCV state is changed as illustrated in Table 16. You will see the abbreviated BCV pair state listed using the SYMCLI commands.

Table 16 BCV pair states

BCV pair state	BCV pair state (abbreviated for display)	Description
Never Established	NeverEstab	The BCV device is available for use, and was never established. Only the BCV device name is valid.
Sync In Progress	SyncInProg	When the Establish action is executed, data is copied from the standard device to the BCV device until both devices contain identical data.
Restore In Progress	RestInProg	When the restore action is executed, data is copied from the BCV to the standard device until both devices contain identical data.
Synchronized	Synchronized	The BCV and standard devices have identical data. Any changes to the standard device are also written to the BCV. The BCV is unavailable to the host for BC processing.
Restored	Restored	The BCV and standard devices have identical data, although the data was originally on the BCV before being synchronized. Any changes to the standard device are also written to the BCV. The BCV is unavailable to the host for BC processing.
Split in Progress	SplitInProg	The BCV devices are in the process of being separated, or split from the standard devices.
Split	Split	The BCV devices are completely separated, or split from the standard devices allowing each device to be accessed separately by the host.
Split No Incremental	SplitNoInc	The BCV devices are completely separated, or split from the standard devices but cannot be incrementally established or restored.
Split Before Sync	SplitBfrSync	The split occurred when a BCV device was synchronizing. The BCV device is separated from the standard device although the BCV device is not completely synchronized.
Split Before Restore	SplitBfrRest	The split occurred when a BCV device was being restored to a standard device. The BCV device is separated from the standard device although the standard device is not completely synchronized.
Invalid	Invalid	Not all metamembers are in the same BCV state.

Transient BCV pair states

When you initially invoke the `symmir` arguments for TimeFinder operations, BCV pairs enter a transient state and upon completion of the action, the BCV pairs enter a final BCV pair state (Table 17).

Table 17 Actions for BCV devices

Argument	Transient state	Final state
`establish`	SyncInProg	Synchronized
`split`	SplitInProg	Split
`restore`	RestInProg	Restored

BCV actions and applicable states

Table 18 describes which BCV control operations can be invoked for a given BCV state.

Invalid states can indicate that the devices in a BCV pair are in a different or *mixed* state.

The `-symforce` option must be used (where noted in the table as F) to force a pair to a specified BCV state.

Table 18 BCV control actions and applicable states

Control Operation	Never Estab	Sync In Prog	Synchronized	Split In Prog	Split	Split No Inc	Split Bfr Sync	Split Bfr Rest	Rest In Prog	Restored	Invalid
`establish -full`	✔				✔	✔	✔[a]	✔[b]			✔
`establish`					✔						
`split`		F[c]	✔						F[c]	✔	
`restore -full`	✔				✔	✔[a]	✔[b]	✔[a]			✔
`restore`					✔						
`attach`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`detach`	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
`cancel`					✔		✔	✔			

a. The BVC must be specified or you must use the `-exact` or `-force` option.

b. The BCV must be specified or you must use both the `-force` and `-symforce` options.

c. The F denotes that you must use the `-symforce` option.

Command options with device groups

Options to the `symmir -g` command line arguments provide more action flexibility to control BCV pairs when you are operating on device(s) of a specified device group. Table 19 lists the `symmir` control operations and the possible options to use when targeting a specified device group.

Table 19 symmir -g control arguments and possible options

Options	Argument action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-bcv	✔	✔	✔	✔	✔	✔	✔	✔
-bypass			✔	✔	✔			
-c, -i	✔	✔	✔	✔	✔	✔	✔	✔
-concurrent	✔	✔						
-consistent			✔					
-exact	✔			✔				
-force, -symforce	✔	✔	✔	✔	✔			✔
-hop2	✔	✔	✔	✔	✔	✔	✔	✔
-instant, -diff			✔					
-noprompt	✔	✔	✔	✔	✔	✔	✔	✔
-not_ready			✔	✔	✔			
-opt	✔							
-ppath			✔					
-preaction, -postaction	✔	✔	✔	✔	✔			
-preservetgtlocks, -lockid	✔	✔	✔	✔	✔			
-protbcvest	✔	✔						
-protect			✔	✔	✔			
-rdb, -dbtype, -db			✔					
-rdf	✔	✔	✔	✔	✔	✔	✔	✔
-remote	✔	✔	✔	✔	✔			
-reverse	✔	✔	✔	✔	✔			
-rrbcv	✔	✔	✔	✔	✔	✔	✔	✔
-star	✔	✔	✔	✔	✔			✔
-std_protect			✔					
-skip	✔	✔	✔					
-v	✔	✔	✔	✔	✔	✔	✔	✔
-vxfs			✔					

Table 20 lists the `symmir` view arguments and the possible options to use when targeting a specified device group.

Table 20 symmir -g view arguments and possible options

Options	Argument action	
	query	verify
-attach	✔	
-bcv, -rrbcv	✔	✔
-bcv_mirrors		✔
-bg	✔	✔
-c, -i	✔	✔
-concurrent		✔
-force		✔
-hop2	✔	✔
-multi	✔	
-offline	✔	✔
-percent	✔	✔
-protbcvest	✔	
-protect	✔	✔
-rdf	✔	✔
-ready		✔
-restinprog		✔
-restored		✔
-sid	✔	✔
-split		✔
-summary	✔	✔
-synched		✔
-syncinprog		✔

Note: The base tasks performed with `symbcv` such as `list`, `associate`, and `disassociate` locally or remotely attached BCV devices, are described at the beginning of this chapter.

Control options

The following sections provide brief descriptions of the various BCV control options.

BCV

The BCV (`-bcv`) option (used in conjunction with the `-rdf` option) indicates that the BCV control operation is targeted at the remote standard mirror of the local RDF1 BCV device and the SRDF-connected BCV device:

```
symmir -g prod -rdf -bcv split
```

Note: "Establishing SRDF-connected BCV pairs" on page 137 contains more information.

BRBCV

The BRBCV (`-brbcv`) option targets the action at the specified remotely-associated RDF BCV devices in the device group.

Both sides

The both sides (`-both_sides`) option performs an instant or consistent split to both the locally and the remotely connected BCV devices (SRDF) within the specified device group.

Note: This option is not supported with R21 groups.

Bypass

The bypass (`-bypass`) option causes the action to bypass host SCSI device reservations.

Concurrent

The concurrent (`-concurrent`) option applies to verify and establish operations. When used in a verify operation, it specifies to verify the standard device and the two most recent BCVs. When used in an establish operation, it specifies to establish a second available BCV.

Count and interval

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

The count (`-c`) option counts the number of times to attempt to acquire exclusive locks on the Symmetrix host database, at the frequency specified by the interval.

If the (`-c`) option is not specified and an interval (`-i`) is specified, your script will loop continuously to display, or until mirroring operations are started.

The interval and count options can be used to continue trying to execute a BCV control command when there is an existing exclusive lock on the Symmetrix host database. If a script invokes a BCV control command, such as establish, on a BCV device pair, an exclusive lock is acquired on the Symmetrix host database. After the establish operation is initiated, the exclusive lock on the Symmetrix host database is released. Given that multiple scripts can invoke BCV control commands at the same time, the count and interval options can be very useful.

For example, to query a BCV pair consisting of standard device `DEV001` and BCV device `BCV001`, every 10 seconds for 1 minute, in the device group `prod`, enter:

```
symmir -g prod query -i 10 -c 6 DEV001
```

Consistent

The consistent (`-consistent`) option can be used to perform consistent split operations using PowerPath-connected devices, the Enginuity Consist feature, or SRDF/A.

Note: For more information, refer to "Consistent split using Enginuity Consistency Assist" on page 169 and "Consistent split for SRDF/A devices" on page 171.

Databases

The following are the database options used with consistent splits:

- The database (`-db`) option used with consistent instant splits specifies a relational database name (not required for Oracle).
- The database type (`-dbtype`) option used with a consistent instant split specifies a relational database type.
- The relational database (`-rdb`) option is for consistent split actions to indicate that the device I/O of the specified database will be frozen just before the instant split is performed, and thawed as soon as the foreground split completes. The host physical devices where the database resides must be PowerPath devices.

Note: "TimeFinder consistent split" on page 167 contains more information.

Device file

The device file (`-file`) option directs the specified operation in the `symmir` command to a device file. The device file contains device pairs (*SymDevnames*) listing a pair per each line (the standard first, a space, and the BCV last within each line entry). Device files can include comment lines that begin with the pound sign (#). The following example illustrates the file format, which specifies three device pairs:

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

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

Exact

The exact (`-exact)` option applies to the full establish or restore operation that causes the standard and BCV device pairing algorithm to select the pairs according to the exact order in which they were added to the specified device group. This option overrides all other pairing options.

Hop2

The hop2 (`-hop2`) option allows you to perform actions on a Symmetrix array two hops away in a Cascaded SRDF configuration.

Force

The force (`-force`) option allows you to override some normal device checking when performing actions on BCVs.

The results of using the `symmir` command with the following arguments, while using the `-force` option, are displayed in Table 21.

Table 21 Results of -force option (page 1 of 2)

Argument	Result
`establish -full`	A standard device will be omitted if none of the BCVs can be paired with it. Allows BCV devices that are in the SplitBfrRestore states.
`establish`	Skip but do not reject devices that are Never Established. Allow devices that are SplitBfrSync.
`split`	Devices in a device group that are not properly synchronized with BCVs, associated with the group, will be omitted. Otherwise, they may be rejected depending on the state of the device.
`restore -full`	Skip but do not reject devices in the group that were Never Established, or are not properly paired and split with BCVs associated with the group. This will also allow BCV devices that are SplitBfrSync to be restored.

Table 21 **Results of -force option (page 2 of 2)**

Argument	Result
`restore`	Skip but do not reject devices in the group that were Never Established, or are not properly paired and split with BCVs associated with the group. This will also allow devices that are in SplitBfrRestore state.
`verify`	Skip but do not reject devices in the group that were Never Established, or are not properly paired and split with BCVs associated with the group.

Instant split

The instant (`-instant`) option improves the performance of a typical split operation by performing a quick foreground BCV split. This option can be made the continual default split mode by setting the following:

```
SYMAPI_DEFAULT_BCV_SPLIT_TYPE = INSTANT
```

in file:

```
/var/symapi/config/options
```

or:

```
C:\Program Files\EMC\symapi\config\options
```

To remove the instant default option, remove the line entry or comment the line out by using a pound sign (#) at the beginning of the line.

Local optimizing

The optimize (`-opt`) option applies to the Full Establish operation that optimizes the disk I/O on the standard/BCV pair selection in the local Symmetrix array to achieve a high copy speed between them. (Basically, the device pair selection attempts to pair devices that are not on the same disk adapter to distribute the I/O.) This option overrides all current pairing relationships.

The following details the sequences of the pairing steps that the optimize option executes:

1. For each STD in a group, the Optimize process searches for a BCV device of the same size where the mirrors of the standard and BCV devices are serviced by separate disk adapters.
2. If no BCV device is found, then the Optimize process searches for a BCV device of the same size where the mirrors of the standard and BCV devices do not share the same physical disk.
3. If no BCV device is found, then the Optimize process searches for any BCV device of the same size.

No prompt

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

Not ready

The not ready (`-not_ready`) option for `restore` and `split` operations sets the target device(s) as Not Ready. Upon completion of a split, each BCV device is set as Not Ready. Upon initiation of a restore, each standard device is set as Not Ready.

Offline

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

PowerPath

The PowerPath (`-ppath`) option used with a consistent instant split applies a list of one or more PowerPath devices to have I/O suspended (where the database information is contained).

Note: "TimeFinder consistent split" on page 167 contains more information.

Pre/Post action

The pre-action (`-preaction`) and post-action (`-postaction`) options apply a script as an action before and after the instant split operation.

Note: "TimeFinder consistent split" on page 167 contains more information.

Protected BCV establish

The protected BCV establish (`-protbcvest`) option is used with the `establish` action to move all mirrors of locally mirrored BCV devices to join the mirrors of a standard device.

Note: Protected BCV establish is not supported if the standard device is a dynamic concurrent SRDF device.

Note: "Performing a protected BCV establish (moving all mirrors)" on page 143 contains more information.

Protected restore

You can restore data from a BCV to a standard device without altering the contents of the BCV, by using the protect (`-protect`) option. It write-disables the BCV mirror(s) during and particularly after the restore operation.

Note: "Protecting BCV data during full or incremental restores" on page 188 contains more information.

RDF

The RDF (`-rdf`) option indicates that the BCV control operation is targeted at the remote mirror of the local RDF standard device and at the SRDF-connected BCV device. For example:

```
symmir -g prod -rdf split
```

When used in conjunction with the `-bcv` option, the BCV control operation is targeted at the remote standard mirror of the local RDF1 BCV device and the SRDF-connected BCV device:

```
symmir -g prod -rdf -bcv split
```

Remote copy

The remote (`-remote`) option functions as a remote data copy flag. It applies to a split operation when the BCV device is a remotely mirrored R1 device, or it applies to an establish or a restore operation when the standard device is a remotely mirrored R2 or R1 device. If this option is not specified, then the mode defaults to inhibit the remote propagation of data to the remote RDF mirror.

Note: "Using the -remote option on multihop split actions" on page 230 contains more information.

Table 22 shows the impact of the remote option (`-remote`) on the RDF pair states.

Table 22 RDF pair states relating to the remote option

RDF pair state prior to a BCV action	RDF pair state without -remote	RDF pair state with -remote
Split	Split	Invalid (write disabled)
Synchronized	Suspended	SyncInProg, then Synchronized when copied
Suspended	Suspended	SyncInProg, then Synchronized when copied
Failed Over	Failed Over	R1 Updated

CAUTION

If you are restoring data from a BCV device to an RDF R2 device, and plan to propagate the same data to the R1 mirror of the R2 device, EMC recommends that you perform this action using the following two commands (instead of using the `-remote` option in one command):

```
symmir -g DgName [-full] restore #restores R2 from its BCV
symrdf -g DgName restore         #restores the R1 from R2
```

Note: Table 52 on page 435 contains more information on the impact of the remote option on SRDF operations.

Remote RBCV

The remote RBCV control (`-rrbcv`) option provides extended BCV control functionality within a device group targeted at the remote mirror of the remotely attached BCV (RBCV) and the remotely attached remote BCV (RRBCV) device.

Reverse

The reverse (`-reverse`) option indicates that the split operation should initiate a reverse data copy from the secondary (fixed) BCV mirror to the first (moving) mirror of the BCV upon completion of the split operation. For example:

```
symmir -g prod -reverse split
```

Note: For more information, refer to "Performing a reverse split" on page 161.

Star

The star (`-star`) option indicates that the action is targeted at devices in STAR mode.

Standard protect

The standard protect (`-std_protect`) option indicates that the split operation should not occur if the standard device will be left in an unprotected state.

Note: For more information, refer to "Performing a Protected split" on page 160.

symforce

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

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

Use care when applying this option.

Symmetrix ID

The Symmetrix ID (`-sid`) option applies the command to a specified Symmetrix ID. (If you set environment variable SYMCLI_SID, this option is not required.) This option with the device filename option allows you to operate on remote BCV pairs in Symmetrix units beyond the first SRDF multihop.

Verbose

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

VERITAS FS

The VERITAS FS (`-vxfs`) option is used with consistent splits that specify a list (for Solaris and HP-UX hosts only) of one or more VERITAS VxFS file system mount points. The file system(s) mounted on this host will be frozen just before the instant split is performed, and thawed as soon as the foreground split completes.

Note: "TimeFinder consistent split" on page 167 contains more information.

Verify options

The following options can be used with the `verify` argument.

BCV mirror

Used with the `verify` argument, the BCV mirror (`-bcv_mirrors`) option verifies that the mirrors of the BCV device(s) are in the indicated state. The default verification is to verify that the mirrors are in the Synchronized state. Alternatively, when the `-syncinprog` or `-restinprog` flag is also specified, then the mirrors of the BCV device(s) are verified against the state that corresponds to the specified flag. For example:

```
symmir -g prod -bcv_mirrors verify
```

Concurrent

Used with the `verify` argument, the concurrent (`-concurrent`) option verifies the synchronization and restore operations of the standard device and the two most recent (concurrent) BCVs.

Restored

Used with the `verify` argument, the restored (`-restored`) option verifies whether the BCV device pair(s) are in the Restored state.

For example, to verify whether the BCV device pair(s) in the device group `prod` are in the Restored state, enter:

```
symmir -g prod -restored verify
```

Restore in progress

Used with the `verify` argument, the restore in progress (`-restinprog`) option verifies whether one or all BCV device pair(s) in a device group are in the RestInProg state:

```
symmir -g prod -restinprog verify
```

Split

Used with the `verify` argument, the split (`-split`) option verifies whether the BCV device pair(s) are only in the Split state. For example:

```
symmir -g prod -split verify
```

Synchronized

Used with the `verify` argument, the synchronized (`-synched`) option verifies whether the BCV device pair(s) are only in the Synchronized state. For example:

```
symmir -g prod -synched verify
```

Sync in progress

Used with the `verify` argument, the sync in progress (`-syncinprog`) option verifies whether one or all BCV device pair(s) in a device group are in the SyncInProg state. For example:

```
symmir -g prod -syncinprog verify
```

Query options

The following options can be used with the `query` argument.

Attach

The attach (`-attach`) option alters the `query` operation to display the BCV preferred attachment information for the standard device(s) in the device group.

Background

The background (`-bg`) option alters a `query` or `verify` operation. For `query` operations, it shows the BCV pairs that are still in the background split mode. For `verify` operations using the `-split` option, it verifies that the BCV pair(s) are in the Split state after completing a background split.

Multi

The multi-BCVs (`-multi`) option directs the `query` operation to display all BCVs paired with the standard device; `-multi` lists BCV devices in chronological order.

Command options with composite groups

Options to the `symmir -cg` command line arguments provide more action flexibility to control BCV pairs when you are operating on device(s) of a specified composite group. Table 23 lists the `symmir` control operations and the possible options to use when targeting a specified composite group.

Table 23 symmir -cg control arguments and possible options (page 1 of 2)

Options	Argument action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-bcv	✔	✔	✔	✔	✔	✔	✔	✔
-bypass			✔	✔	✔			
-c, -i	✔	✔	✔	✔	✔	✔	✔	✔
-concurrent	✔	✔						
-consistent, -both_sides			✔					
-exact	✔			✔				
-force, -symforce	✔	✔	✔	✔	✔			
-hop2	✔	✔	✔	✔	✔	✔	✔	✔
-instant, -diff			✔					
-noprompt	✔	✔	✔	✔	✔	✔	✔	✔
-not_ready			✔	✔	✔			
-opt	✔							
-opt_rag	✔							
-ppath			✔					
-preaction, -postaction	✔	✔	✔	✔	✔			
-protbcvest	✔	✔						
-protect			✔	✔	✔			
-rdb, -dbtype, -db			✔					
-rdf	✔	✔	✔	✔	✔	✔	✔	✔
-remote	✔	✔	✔	✔	✔			
-reverse	✔	✔	✔	✔	✔			
-rrbcv	✔	✔	✔	✔	✔	✔	✔	✔
-sid	✔	✔	✔	✔	✔	✔	✔	✔
-skip	✔	✔	✔					
-star	✔	✔	✔	✔	✔			

Table 23 symmir -cg control arguments and possible options (page 2 of 2)

Options	Argument action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-std_protect			✔					
-v	✔	✔	✔	✔	✔	✔	✔	✔
-vxfs			✔					

Table 24 lists the `symmir` view arguments and the possible options to use when targeting a specified composite group.

Table 24 symmir -cg view arguments and possible options

Options	Argument Action	
	query	verify
-attach	✔	
-bcv, -rrbcv	✔	✔
-bcv_mirrors		✔
-bg	✔	✔
-c, -i	✔	✔
-concurrent		✔
-hop2	✔	✔
-force		✔
-multi	✔	
-offline	✔	✔
-percent	✔	✔
-protect	✔	
-rdf	✔	✔
-restored		✔
-sid	✔	✔
-split		✔
-synched		✔
-syncinprog		✔
-ready		✔
-restinprog		✔
-protbcvest	✔	
-summary	✔	

Remote optimizing option

The remote optimize (-opt_rag) option only applies to the full Establish operation in a remote Symmetrix array that optimizes the disk I/O on the standard/BCV pair selection to achieve a high copy speed between them. (Basically, the device pair selection attempts to pair devices that are not on the same disk adapter to distribute I/O.) This option overrides all current pairing relationships.

Note: This option is only applicable for remote Symmetrix array optimization targeting composite groups (-cg).

The command line must include the -rdf option, as follows:

```
symmir -cg CgName -full establish -rdf -opt_rag
```

Other options

For local optimization, use the -opt option (refer to "Local optimizing" on page 217).

For all other remaining option descriptions, refer to "Control options" on page 215.

Command options with device files

With the `symmir -file` command, you can perform similar control operations on BCV device pairs defined in a device file of a specified Symmetrix array as you can when directing `symmir` to device groups (`-g`). These control operations (arguments) have similar options that allow flexibility in controlling STD/BCV pairs defined in a device file, as opposed to a device group. This command is particularly useful when operating on RDF BCV pairs in a remote Symmetrix array in the second-level multihop SRDF link.

Note: "Device file" on page 216 contains information on creating a device file.

Table 25 lists the `symmir` control operations and the possible options to use when targeting pairs specified in a device file of a given Symmetrix array.

Table 25 symmir -file control arguments and possible options

Options	Argument Action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
`-bypass`			✔	✔	✔			
`-c`, `-i`	✔	✔	✔	✔	✔	✔	✔	✔
`-consistent`			✔					
`-force`, `-symforce`	✔	✔	✔	✔	✔			✔
`-instant`, `-diff`			✔					
`-noprompt`	✔	✔	✔	✔	✔	✔	✔	✔
`-not_ready`			✔	✔	✔			
`-ppath`			✔					
`-preaction`, `-postaction`	✔	✔	✔	✔	✔			
`-preservetgtlocks, -lockid`	✔	✔	✔	✔	✔			
`-protbcvest`	✔	✔						
`-protect`			✔	✔	✔			
`-rdb`, `-dbtype`, `-db`			✔					
`-remote`	✔	✔	✔	✔	✔			
`-reverse`	✔	✔	✔	✔	✔			
`-skip`	✔	✔	✔					
`-star`	✔	✔	✔	✔	✔			✔
`-std_protect`			✔					
`-v`	✔	✔	✔	✔	✔	✔	✔	✔
`-vxfs`			✔					

Table 26 lists the `symmir` view arguments and the possible options to use when targeting pairs specified in a device file of a given Symmetrix array.

Table 26 symmir -file view arguments and possible options

Options	Argument Action	
	query	verify
`-attach`	✔	
`-bcv_mirrors`		✔
`-bg`	✔	✔
`-c`, `-i`	✔	✔
`-concurrent`		✔
`-force`	✔	✔
`-multi`	✔	
`-offline`	✔	✔
`-percent`	✔	✔
`-protbcvest`	✔	
`-protect`	✔	
`-ready`		✔
`-restinprog`		✔
`-restored`		✔
`-split`		✔
`-summary`	✔	
`-synched`		✔
`-syncinprog`		✔

Note: The Symmetrix ID option (`-sid`) is required for all `symmir -file` commands.

Various remote multihop configurations

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

As Figure 92 on page 229 shows, you can have multiple sites (for example, remote sites C, E, F, and H) on SRDF links to remotely mirror a local Symmetrix array at site A. Remote site F, functioning as a remote mirror to the standard devices at site A, is most typical. You then can have a third site on an SRDF link (remote site H) to remotely mirror just the BCV devices in the Symmetrix array at site A.

You can also multihop to a second level SRDF where Remote site G functions as a remote mirror to the standard devices of site A and Remote site I remotely mirrors Site A's BCV.

In addition, you can also create a Cascaded SRDF configuration, where tertiary site B functions as a remote partner to the R21 device at Site C, which is the remote partner of the local RDF standard device at Site A; and tertiary site D functions as a remote partner to the R21 device at Site E, which is the remote partner of the local BCV device at Site A.

Command `symmir` manages each of the BCV pairs at any site while `symrdf` manages the SRDF pairs in the SRDF link.

System-wide device groups

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

```
symdg create prod -type RDF1
symld -g prod add dev 0001 -sid 0001 DEV001
symbcv -g prod associate dev 000A BCV001
symbcv -g prod associate dev 000C -rdf RBCV001
symbcv -g prod associate dev 0009 -bcv -rdf BRBCV001
symbcv -g prod associate dev 0004 -rrdf RRBCV001
symbcv -g prod associate dev 0004 -hop2 2BCV001
```

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

Commands to various multihop devices and links

This section describes the command application of targeting the various devices and links in complex multihop SRDF environments.

The following sequence of commands steps through some basic control operations that touch every device and RDF link in a complex multihop configuration. The following numbering of commands directly associates with the bubble numbers shown in the Figure 92 on page 229.

1	`symrdf -g <> establish`	Creates the standard-associated hop 1 copy.
2	`symmir -g <> split -rdf`	Splits the standard-associated hop 1 BCV device pair.
3	`symrdf -g <> establish -rbcv`	Creates the standard-associated hop 2 copy.
4	`symrdf -g <> restore -rbcv`	Restores the standard-associated hop 1 BCV with the hop 2 copy.
5	`symmir -g <> restore -rdf`	Restores the standard-associated hop 1 copy with the hop 1 BCV.
6	`symrdf -g <> restore`	Restores the standard device with the hop 1 copy.
7	`symmir -g <> split`	Splits the standard/BCV pair.
8	`symrdf -g <> establish -bcv`	Creates the BCV-associated hop 1 remote copy.
9	`symmir -g <> split -rdf -bcv`	Splits the BCV-associated hop 1 device pair.
10	`symrdf -g <> establish -brbcv`	Creates the BCV-associated hop 2 copy.
11	`symrdf -g <> restore -brbcv`	Restores the BCV-associated hop 1 BCV with the hop 2 copy.
12	`symmir -g <> restore -rdf -bcv`	Restores the BCV-associated hop 1 copy with the hop 1 BCV.
13	`symrdf -g <> restore -bcv`	Restores the BCV device with the hop 1 copy.
14	`symmir -g <> restore`	Restores the standard device with the BCV copy.
15	`symmir -file <> -sid 044 establish`	Creates the BCV-associated with the remote partner of the BRBCV (file only).
16	`symmir -file <> -sid 044 split`	Splits the BCV-associated with the remote partner of the BRBCV (file only).
17	`symrdf -g <> establish`	Creates the standard-associated hop 1 copy.
18	`symrdf -g <> establish -hop2`	Creates the standard-associated hop 2 copy.
19	`symmir -g <> establish -hop2`	Creates the BCV-associated hop 2 BCV copy.
20	`symmir -g <> split -hop2`	Splits the BCV-associated hop 2 device pair.
21	`symrdf -g <> restore -hop2`	Restores the standard with the hop 2 copy.
22	`symrdf -g <> restore`	Restores the standard device with the hop 1 copy.
23	`symrdf -g <> establish -bcv`	Creates the BCV-associated hop 1 remote copy.
24	`symrdf -g <> establish -bcv -hop2`	Creates the BCV-associated hop 2 copy.
25	`symrdf -g <> restore -bcv -hop2`	Restores the BCV-associated hop 2 copy.
26	`symrdf -g <> restore -bcv`	Restores the BCV device with the hop 1 copy.

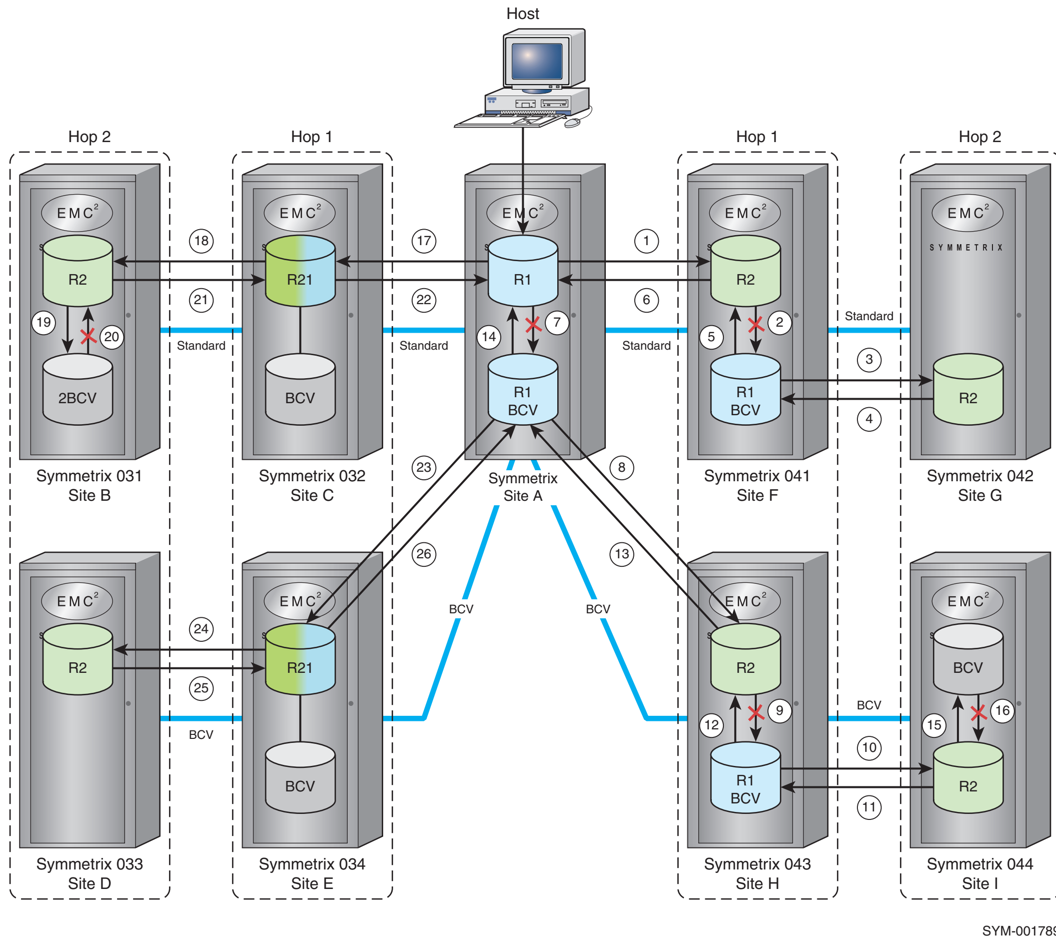

Figure 92 Control operations on multihop SRDF configurations

Second-level controls for multihop SRDF environments

As previously described, second-level multihop control operations were accomplished using the device file (`-file`) option for managing RDF BCV device pairs. Since Solutions Enabler Version 5.3, you can use the `symmir` command for device groups (`-g`) for BCV control capability in second-level multihop SRDF environments.

The remote RDF BCV (RRBCV) devices must have been previously associated with the device group using the `symbcv -rrdf` command. "Associating devices in a compounded remote configuration" on page 126 contains specific information about how to associate second-level multihop BCVs with a device group.

Once the RRBCV devices have been associated with the device group, you can use the `symmir` command with the `-rrbcv` option to perform control operations on the remote mirror of the remote BCV to become established, split, or restored from its BCV. Other second-level multihop BCV control operations available with the `symmir` command include `query`, `verify`, `attach`, `detach`, and `cancel`.

Note: Since Solutions Enabler Version 5.4, the `symmir` command and the `-rrbcv` option have been extended to work for composite groups (`-cg`).

"Establishing second-level remote BCV pairs" on page 139 provides an example of a BCV establish operation in a second-level multihop SRDF environment.

Using the -remote option on multihop split actions

This section describes the command application of targeting the various devices and links with the `-remote` option in complex multihop SRDF environments.

The following sequence of commands steps through some basic control operations that touch every device and RDF link in a complex multihop configuration. The following numbering of commands directly associates with the callouts shown in the Figure 93 on page 231.

1	`symrdf -g <> establish`	Creates the standard-associated hop 1 copy.
2	`symmir -g <> split -rdf -remote`	Splits the standard-associated hop 1 BCV device pair and creates a standard-associated hop 2 copy of the hop 1 BCV.
3	`symmir -g <> split -remote`	Splits the standard/BCV pair and creates a BCV-associated hop 1 copy of the local BCV.
4	`symmir -g <> split -rdf -bcv -remote`	Splits the BCV-associated hop 1 BCV device pair and creates a BCV-associated hop 2 copy of the hop 1 BCV.
5	`symmir -g <> split -rrbcv`	Splits the BCV-associated hop 2 BCV device pair. You cannot use the `-remote` option here.

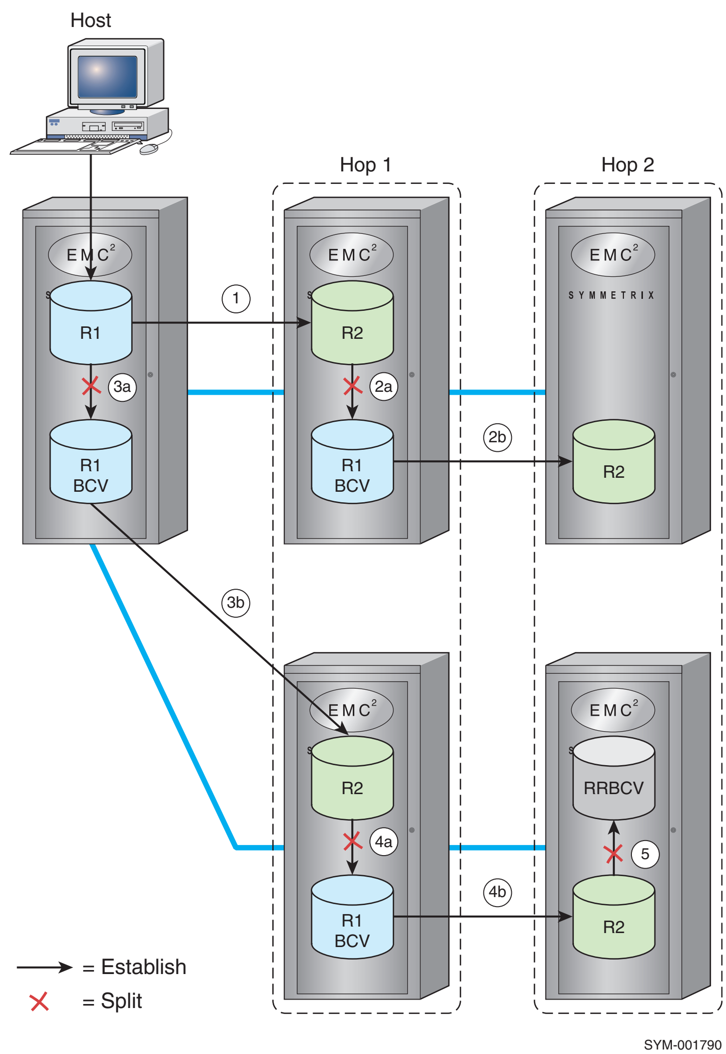

Figure 93 **The -remote option on multihop configurations**

Relabeling devices for Windows 2000

With Solutions Enabler SYMCLI, Windows system administrators can relabel devices.

In certain situations, relabeling devices is required when devices are under a MS Windows-type volume manager's control. If a BCV (TimeFinder) device holds an identical copy of its standard (paired) device, when the BCV device becomes ready to the operating system, the volume manager will detect two identical volumes with different mount points. This can cause the volume manager to exit and crash a system.

Depending on the current state of the device, up to five different SYMCLI commands may need to be executed for a relabel device operation:

`symdg, symld, symbcv, symlabel, symmir`

Note: The *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide* contains information on the first four commands.

The following sequence of command steps are an example of a relabel operation:

1. If these devices are currently under TimeFinder control and are part of a disk group, then skip to step 4.
2. To create a group and add both your STD and BCV devices, enter:

   ```
   symdg create -type REGULAR my_group
   ```
3. Add both your STD (standard) and BCV (TimeFinder) devices:

   ```
   symld  add       dev 0001 -g my_group
   symbcv associate dev 0002 -g my_group
   ```
4. Define a label on both your STD and BCV devices. You are required to enter an 8-character hexadecimal value:

   ```
   symlabel -g my_group -type WNT define DEV001 LABEL 11111111
   symlabel -g my_group -type WNT define BCV001 LABEL 22222222 -bcv
   ```

   Note: Note that specifying the actual label value is optional. If you omit the label, SYMCLI will find the device label for the command execution.
5. If these devices are currently established or were previously fully established under TimeFinder control, then proceed to step 7.
6. Now perform a full establish of your newly created group of two devices:

   ```
   symmir -g my_group -full establish
   ```
7. Once the TimeFinder pair is synchronized, split the pair:

   ```
   symmir -g my_group split -not_ready
   ```

   This step places the BCV devices in the NR (Not Ready) state after the split.
8. Relabel the BCV devices:

   ```
   symld -g my_group relabel -bcv
   ```

   This step relabels the BCV devices.
9. Change the status of the BCV devices from NR (Not Ready) to RW (Ready):

   ```
   symld -g my_group ready -bcv
   ```

10. To relabel the standard devices after a **restore** operation, enter:

```
symmir -g my_group restore -not_ready
symld -g my_group relabel !Relabels the STD devices
symld -g my_group ready  !Changes the STD device status to ready
```

PART 2

Operational Examples

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

Chapter 5, “Performing TimeFinder/Clone Operations,” provides examples on cloning volume data on multiple target devices from a single source device.

Chapter 6, “Performing TimeFinder/Snap Operations,” provides examples on creating space-saving, instant point-in-time copies of volume data on multiple target devices from a single source device.

Chapter 7, “Attaching, Querying, and Verifying with TimeFinder commands,” provides examples on using the attach, query, and verify operations with TimeFinder family products.

Chapter 8, “Setting Up TimeFinder/Mirror BCV Pairs,” provides examples on setting up BCV pairs in the TimeFinder/Mirror environment. It focuses on creating and populating a device group and creating a BCV pair from a standard and BCV that belong to the same device group.

Chapter 9, “Setting Up TimeFinder/Mirror Multiple and Concurrent BCVs,” provides examples on setting up multiple and concurrent BCVs in the TimeFinder/Mirror environment. It focuses on pairing one standard device sequentially with multiple BCVs and pairing one standard device simultaneously with two BCVs.

Chapter 10, “Performing TimeFinder/Mirror Control Operations,” provides examples on creating and splitting a BCV pair, retrieving original BCV data from a fixed BCV mirror, and performing a BCV action while making the BCV or the standard device temporarily inaccessible to the host.

Chapter 11, “Performing Consistent Splits with TimeFinder Family,” provides examples on splitting off a consistent, DBMS-restartable BCV copy of your database without having to quiesce or shut down the database first.

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

5

Performing TimeFinder/Clone Operations

This chapter provides examples on cloning volume data on multiple target devices from a single source device.

The following examples illustrate TimeFinder/Clone functionality in Solutions Enabler releases up through Version 7.0 running on Symmetrix arrays configured with Enginuity 5670 and earlier, 5671, and 5874.

Example 1: Cloning a copy of a standard device to other standard devices

The hardware setup consists of a single Symmetrix array (sid 041) connected to a Solaris controlling host. The source of the clone is standard device 0101. The targets are standard devices 0102 and 0103.

- The `symdg` command creates a Regular type device group named clonegrp.

  `symdg create clonegrp -type regular`

- The `symld` command adds a device range (0101 through 0103) to the device group. By default, their logical device names will be DEV001, DEV002, and DEV003, respectively.

  `symld -g clonegrp addall dev -range 101:103`

- The `symclone create` command, by default, creates a CopyOnAccess copy session so that selected data can be cloned from source device DEV001 to target DEV002 when the clone operation is activated.

```
symclone -g clonegrp create DEV001 sym ld DEV002 -noprompt

'Create' operation execution is in progress for device 'DEV001' paired with
target device 'DEV002' in device group 'clonegrp'. Please wait...

'Create' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV002'.
```

- The `symclone query` command indicates that the clone pair is in the Created state. The CG columns relate to the Legend: an X in the column means that a Legend item applies to the clone pair; a dot (.) in the column means that item does not apply. Only (G) in the Legend applies to this clone pair.

  `symclone -g clonegrp query DEV001`

```
Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device           Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym CG SRC <=> TGT   (%)
------------------------- ------------------- ------------ ----

DEV001     0101     138090 DEV002       0102 .X Created        0

Total            --------
  MB(s)             4315.3

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone activate` command activates the clone operation.

```
symclone -g clonegrp activate DEV001 sym ld DEV002 -noprompt

'Activate' operation execution is in progress for device 'DEV001' paired with target
device 'DEV002' in device group 'clonegrp'. Please wait...

'Activate' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV002'.
```

- The `symclone query` command indicates that the clone pair is now in the CopyOnAccess state.

```
symclone -g clonegrp query DEV001

Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State        Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym   CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001     0101     138090 DEV002       0102 .X CopyOnAccess   0

Total              --------
  MB(s)              4315.3

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- To ensure that you do not copy over previously copied data, the clone operation results in putting a hold on the target device. The `symld list` command with the `-held` option displays any devices in the device group that are currently being held. This hold is removed when you terminate the clone pair session.

```
symld -g clonegrp -held list

Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

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

DEV002      rdmp/c4t0d161s2    0102   01C:0 16B:C2  2-Way Mir          RW    4315
```

- The following `symclone create` command creates a copy session between the same source device (DEV001) and a different target device (DEV003). The `-copy` option will cause the *immediate* initiation of a *full* copy between the devices *when the copy session is activated*.

```
symclone -g clonegrp create DEV001 sym ld DEV003 -copy -noprompt

'Create' operation execution is in progress for device 'DEV001' paired with
target device 'DEV003' in device group 'clonegrp'. Please wait...

'Create' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV003'.
```

- A `symclone query` command with the `-multi` option displays the multiple copy sessions for source device DEV001 and the current state of each clone pair (DEV001/DEV003 and DEV001/DEV002).

```
symclone -g clonegrp query DEV001 -multi

Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device            Target Device         State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001     0101     138090 DEV003       0103 XX Created         0
                    138090 DEV002       0102 .X CopyOnAccess    0

Total                --------
  MB(s)               4315.3

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone activate` command initiates the full copy from DEV001 to DEV003.

```
symclone -g clonegrp activate DEV001 sym ld DEV003 -noprompt

'Activate' operation execution is in progress for device 'DEV001' paired with
target device 'DEV003' in device group 'clonegrp'. Please wait...

'Activate' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV003'.
```

- Another `symclone query` displays again the multiple copy sessions for source device DEV001 and an updated state for each clone pair. The state of the DEV001/DEV003 pair has changed from Created to CopyInProg (which is 1 % complete). The state of the DEV001/DEV002 pair is still CopyOnAccess.

```
symclone -g clonegrp query DEV001 -multi

Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001    0101     136639 DEV003      0103 XX CopyInProg     1
                   138090 DEV002      0102 .X CopyOnAccess   0

Total              --------
  MB(s)              4270.0

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The following `symclone terminate` command ends the DEV001/DEV002 clone pair session normally. Normal termination is possible whenever a clone pair is in the Created, Copied, or CopyOnAccess state.

```
symclone -g clonegrp terminate DEV001 sym ld DEV002 -noprompt

'Terminate' operation execution is in progress for device 'DEV001' paired with
target device 'DEV002' in device group 'clonegrp'. Please wait...

'Terminate' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV002'.
```

- The following `symclone terminate` command attempts to end the DEV001/DEV003 clone pair session normally, but cannot do so because the DEV001/DEV003 pair state is CopyInProg.

```
symclone -g clonegrp terminate DEV001 sym ld DEV003 -noprompt

'Terminate' operation execution is in progress for device 'DEV001' paired
with target device 'DEV003' in device group 'clonegrp'. Please wait...

The session cannot be terminated because a copy is in progress unless the
Symmetrix force flag is used
```

- The following `symclone terminate` command uses the `-symforce` option to force an end to the DEV001/DEV003 clone pair session. If you need to use the `-symforce` option, you should unmount file systems and shut down applications and databases prior to issuing this command.

```
symclone -g clonegrp terminate DEV001 sym ld DEV003 -noprompt -symforce

'Terminate' operation execution is in progress for device 'DEV001' paired with
target device 'DEV003' in device group 'clonegrp'. Please wait...

'Terminate' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV003'.
```

Example 2: Cloning a copy of a standard device to BCV devices

In this example, the three standard devices (0101, 0102, and 0103) currently in the device group from Example 1 are all source devices for this clone operation. Three BCV devices (0215, 0216, and 0217) are added as targets.

- The following command associates three BCV devices (0215, 0216, and 0217) with the device group.

  `symbcv -g clonegrp associateall dev -range 215:217`

- The `symmir query` command checks the device group for any current copy sessions. Because all copy sessions for devices in the device group were stopped in Example 1, the output below indicates no clone devices in the group.

  `symclone -g clonegrp query`

  ```
  Device group 'clonegrp' does not have any devices that are Clone
  source devices
  ```

- The following command, by default, creates CopyOnAccess copy sessions for the standards and BCVs, pairing them in the order that they were added to the device group (for example, the first standard device, 0101, with the first BCV device, 0215).

`symclone -g clonegrp create -v -noprompt`

```
'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

SELECTING Source devices in the group:

  Device: 0101 [SELECTED]
  Device: 0102 [SELECTED]
  Device: 0103 [SELECTED]

SELECTING Target devices in the group:

  Device: 0215 [SELECTED]
  Device: 0216 [SELECTED]
  Device: 0217 [SELECTED]

PAIRING of Source and Target devices:

  Devices: 0101(S) - 0215(T) [PAIRED]
  Devices: 0102(S) - 0216(T) [PAIRED]
  Devices: 0103(S) - 0217(T) [PAIRED]

STARTING a Clone 'CREATE' operation.

The Clone 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for device group 'clonegrp'.
```

- The following query displays the clone pairs and the state of each (Created).

symclone -g clonegrp query -multi

```
Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                 Protected
Logical    Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001    0101     138090 BCV001      0215 .X Created        0
DEV002    0102     138090 BCV002      0216 .X Created        0
DEV003    0103     138090 BCV003      0217 .X Created        0

Total             --------
  Track(s)          414270
  MB(s)            12945.9

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone activate` command activates the clone operation for all clone pairs in the device group.

symclone -g clonegrp activate -noprompt

```
'Activate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.
```

- Another query displays the clone pairs and changed state of each (from Created to CopyOnAccess).

symclone -g clonegrp query -multi

```
Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                 Protected
Logical    Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001    0101     138090 BCV001      0215 .X CopyOnAccess   0
DEV002    0102     138090 BCV002      0216 .X CopyOnAccess   0
DEV003    0103     138090 BCV003      0217 .X CopyOnAccess   0
Total             --------
  Track(s)          414270
  MB(s)            12945.9

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The following `symclone terminate` command ends the clone pair sessions of all three clone pairs in the device group.

```
symclone -g clonegrp terminate -noprompt

'Terminate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.
```

- The `symclone query` command with the `-multi` option checks the device group for any current copy sessions. Because all copy sessions for devices in the device group were stopped, the output below confirms that there are no longer any clone devices in the group.

```
symclone -g clonegrp query -multi

Device group 'clonegrp' does not have any devices that are Clone
source devices
```

- The `symclone create` command creates three new clone pair sessions with the same standards and BCVs. However, this time, the `-copy` option is used to set up the clone operation for *immediate* copying of an entire source device instead of the deferred clone operation of CopyOnAccess.

```
symclone -g clonegrp create -copy -noprompt

'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Create' operation successfully executed for device group 'clonegrp'.
```

- This `symclone activate` command also includes `-preaction` and `-postaction` options to run scripts named pre.sh and post.sh, respectively. An example of a preaction script would be one that unmounts a file system on the target devices; a postaction script might be one that mounts the file system on the target devices.

```
symclone -g clonegrp activate -noprompt -preaction ./pre.sh
                                         -postaction ./post.sh

'Activate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.
```

- The following query shows the state of the clone pairs (CopyInProg).

symclone -g clonegrp query -multi

```
Device Group (DG) Name: clonegrp
DG's Type            : REGULAR
DG's Symmetrix ID    : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical     Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001     0101     134341 BCV001      0215 XX CopyInProg     2
DEV002     0102     135729 BCV002      0216 XX CopyInProg     1
DEV003     0103     134205 BCV003      0217 XX CopyInProg     2

Total            --------
  Track(s)         404275
  MB(s)           12633.6

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone verify` command with the `-copied` option checks all clone pairs every 120 seconds until all are in the Copied state.

symclone -g clonegrp verify -copied -i 120

```
None of the devices in group 'clonegrp' are in the 'Copied' state.

None of the devices in group 'clonegrp' are in the 'Copied' state.

All of the devices in group 'clonegrp' are in the 'Copied' state.
```

- One `symclone terminate` command can stop multiple copy sessions. Stopping a copy session deletes clone pair information from the SYMAPI database and removes any hold on a target device.

symclone -g clonegrp terminate -noprompt DEV001 sym ld BCV001
DEV002 sym ld BCV002
DEV003 sym ld BCV003

```
'Terminate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.
```

- The `symclone create` command with the `-optimize` option uses optimization to create clone pairs from the devices in the device group. Optimization attempts to match devices in such as way as to achieve a high-speed copy between them, first trying to pair devices on separate disk adapters (DA) and, failing that, devices on the same disk adapter but on different SCSI interfaces of the disk adapter and, failing that, devices that may be on the same DA and SCSI interface but on a separate spindle.

```
symclone -g clonegrp create -optimize -v -noprompt

'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

SELECTING Source devices in the group:

  Device: 0101 [SELECTED]
  Device: 0102 [SELECTED]
  Device: 0103 [SELECTED]

SELECTING Target devices in the group:

  Device: 0215 [SELECTED]
  Device: 0216 [SELECTED]
  Device: 0217 [SELECTED]

PAIRING of Source and Target devices:

  Devices: 0101(S) - 0216(T) [PAIRED]
  Devices: 0102(S) - 0215(T) [PAIRED]
  Devices: 0103(S) - 0217(T) [PAIRED]

STARTING a Clone 'CREATE' operation.

The Clone 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for device group 'clonegrp'.
```

- The following query displays the new clone pairs. Optimization matched two of the three pairs differently than the previous clone operation: DEV001 with BCV002, and DEV002 with BCV001.

```
symclone -g clonegrp query -multi

Device Group (DG) Name: clonegrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym  Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001     0101     138090 BCV002      0216 .X Created        0
DEV002     0102     138090 BCV001      0215 .X Created        0
DEV003     0103     138090 BCV003      0217 .X Created        0
Total              --------
  Track(s)          414270
  MB(s)            12945.9

Legend:
(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone activate` command activates the clone operation for all clone pairs in the device group.

```
symclone -g clonegrp activate -noprompt

'Activate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.
```

- The `symclone verify` command with the `-copyonaccess` option checks all clone pairs every 120 seconds until all are in the CopyOnAccess state.

```
symclone -g clonegrp verify -copyonaccess -i 120

All devices in group 'clonegrp' have a paired TGT device that is in the
'CopyOnAccess' state.
```

- The following `symclone terminate` command initiates termination of the three copy sessions.

```
symclone -g clonegrp terminate -noprompt

'Terminate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.
```

- The `symclone list` command indicates that the copy sessions were successfully terminated and no longer exist.

```
symclone list

Symmetrix ID: 000187900041

No Copy sessions found
```

Example 3: Cloning copies from a BCV source device

This example creates a new device group and pairs Symmetrix standard device 0101 with BCV 0215 as a BCV pair. The source of the clone is BCV 0215. The target is standard device 0102.

- The `symdg` command creates a device group named `clonebcv`. The `symld` commands add two standard devices to the group. By default, the logical device name of device 0101 is DEV001. The example assigns CLONETGT as the logical device name for device 0102. The `symbcv` command associates one BCV with the group. The example assigns CLONESRC as the BCV's logical device name.

```
symdg create clonebcv
symld -g clonebcv add dev 101
symld -g clonebcv add dev 102 CLONETGT
symbcv -g clonebcv associate dev 215 CLONESRC
```

- The `symdg show` command displays the details of the device group. The logical device names CLONETGT and CLONESRC make it easier to identify the clone target and clone source.

```
symdg show clonebcv

Group Name:  clonebcv

    Group Type                                   : REGULAR
    Valid                                        : Yes
    Symmetrix ID                                 : 000187900041
    Group Creation Time                          : Tue Mar 18 14:15:31 2003
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :    2
    Number of Associated GK's                    :    0
    Number of Locally-associated BCV's           :    1
    Number of Locally-associated VDEV's          :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0

    Standard (STD) Devices (2):
        {
        --------------------------------------------------------------------
                                                      Sym               Cap
        LdevName              PdevName                Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/vx/rdmp/c4t0d160s2 0101      RW      4315
        CLONETGT              /dev/vx/rdmp/c4t0d161s2 0102      RW      4315
        }

    BCV Devices Locally-associated (1):
        {
        --------------------------------------------------------------------
                                                      Sym               Cap
        LdevName              PdevName                Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        CLONESRC              N/A                     0215      RW      4315
        }
```

- The `symmir establish` command initiates a full establish operation on the BCV pair identified by their logical device names (standard device DEV001 with the BCV named CLONESRC).

```
symmir -g clonebcv establish -full DEV001 bcv ld CLONESRC -noprompt

'Full Establish' operation execution is in progress for device 'DEV001' paired with
BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in group
'clonebcv' paired with BCV device 'CLONESRC'.
```

- The `symmir verify` command checks the establish operation every 30 seconds and verifies when the operation is complete. That is, DEV001 has been fully copied to CLONESRC.

```
symmir -g clonebcv verify DEV001 -i 30

Device 'DEV001' in group 'clonebcv' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'clonebcv' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'clonebcv' is in the 'Synchronized or Restored' state.
```

- The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation.

```
symmir -g clonebcv split DEV001 -instant -not_ready -noprompt

'Split' operation execution is in progress for device 'DEV001' in device group
'clonebcv'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'clonebcv'.
```

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

```
symmir -g clonebcv verify DEV001 -split -bg

Device 'DEV001' in group 'clonebcv' has finished splitting in the background.
```

- The following command, by default, creates a CopyOnAccess copy session for source device CLONESRC and target device CLONETGT.

```
symclone -g clonebcv create CLONESRC sym ld CLONETGT -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'clonebcv'
paired with target device 'CLONETGT'.
```

- The following query displays the new clone pair and its state (Created).

symclone -g clonebcv query

```
Device Group (DG) Name: clonebcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device           Target Device       State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym   CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

CLONESRC  0215     138090 CLONETGT    0102 .X Created        0

Total            --------
  MB(s)            4315.3

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symdev show` command displays the details of device 0215 (the CLONESRC device).

symdev show 215

```
Symmetrix ID: 000187900041

    Device Physical Name     : Not Visible

    Device Symmetrix Name    : 0215
    Device Serial ID         : N/A
    Symmetrix ID             : 000187900041

    Device Group Name        : clonebcv
    Device Logical Name      : CLONESRC

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

    Device Emulation Type    : FBA
    Device Defined Label Type: N/A
    Device Defined Label     : N/A
    Device Sub System Id     : 0x0142

    Device Block Size        : 512
    Device Capacity
        {
        Cylinders            :       9206
        Tracks               :     138090
        512-byte Blocks      :    8837760
        MegaBytes            :       4315
        KiloBytes            :    4418880
        }

   Device Configuration     : 2-Way BCV Mir   (Non-Exclusive Access)

    Device is WORM Enabled   : No
    Device is WORM Protected : No

    SCSI-3 Persistent Reserve: Disabled

    Dynamic Spare Invoked    : No
```

```
Dynamic RDF Capability    : None

Device Service State      : Normal

Device Status             : Not Ready        (NR)
Device SA Status          : N/A              (N/A)

Mirror Set Type           : [Data,Data,N/A,N/A]

Mirror Set DA Status      : [RW,RW,N/A,N/A]

Mirror Set Inv. Tracks    : [0,112458,0,0]

Back End Disk Director Information
    {
    Hyper Type                                : Data
    Hyper Status                            : Ready           (RW)
    Disk [Director, Interface, TID]           : [01A, C, 4]
    Disk Director Volume Number               :   61
    Hyper Number                              :    9
    Disk Capacity                             : 69879m

    Hyper Type                                : Data
    Hyper Status                            : Ready           (RW)
    Disk [Director, Interface, TID]           : [15A, C, 4]
    Disk Director Volume Number               :   61
    Hyper Number                              :    9
    Disk Capacity                             : 69879m
    }
BCV Pair Information
    {
    Standard (STD) Device Symmetrix Name      : 0101
    Standard (STD) Device Serial ID           : 41101330
    Standard (STD) Device Group Name          : clonebcv

    BCV Device Symmetrix Name                 : 0215
    BCV Device Serial ID                      : Not Visible
    BCV Device Associated Group Name          : clonebcv

   BCV Device Status                        : Not Ready      (NR BCV)

    State of Pair ( STD <=\=> BCV )           : Split
   Time of Last BCV Action                 : Tue Mar 18 14:28:02 2003

    State of BCV Mirrors                      : SyncInProg

    BCV State Flags                           : (AllReady)
    Percent Split                             : 100%

    Number of Inv. Tracks for STD Device      :        0
    Number of Inv. Tracks for BCV Device      :       30
    }

Clone Device Information
    {
    Source (SRC) Device Symmetrix Name        : 0215
    Source (SRC) Device Group Name            : clonebcv

    Target (TGT) Device Symmetrix Name        : 0102
    Target (TGT) Device Group Name            : clonebcv

    State of Session ( SRC ==> TGT )          : Created
    Percent Copied                            : 0%

   Time of Last Clone Action               : Tue Mar 18 14:29:14 2003
```

```
Clone State Flags                             : None

Number of Prot. Tracks for SRC Device   : 138090
Number of Indir Tracks for TGT Device   : 138090

}
```

- The `symclone activate` command activates the clone operation for the clone pair.

symclone -g clonebcv activate CLONESRC sym ld CLONETGT -noprompt

```
'Activate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Activate' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

- The `symclone query` command displays the state of the clone operation (CopyOnAccess).

symclone -g clonebcv query

```
Device Group (DG) Name: clonebcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device          Target Device       State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym  Tracks     Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

CLONESRC  0215     138090 CLONETGT    0102 .X CopyOnAccess   0

Total            --------
  MB(s)            4315.3

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The following `symclone terminate` command stops the copy session. The verbose (`-v`) option provides a more detailed output.

symclone -g clonebcv terminate CLONESRC sym ld CLONETGT -noprompt -v

```
'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT' in device group 'clonebcv'. Please wait...

PAIRING of Source and Target devices:

 Devices: 0215(S) - 0102(T) [PAIRED]

STARTING a Clone 'TERMINATE' operation.

The Clone 'TERMINATE' operation SUCCEEDED.

'Terminate' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

- The `symmir query` command displays that the copy session was stopped by indicating that devices in the device group are no longer listed in the SYMAPI database as clone devices.

```
symclone -g clonebcv query

Device group 'clonebcv' does not have any devices that are Clone
source devices
```

- The `symld list` command with the `-held` option displays that the hold on the target device was released by indicating that no devices in the device group are being held.

```
symld -g clonebcv -held list

Device Group (DG) Name: clonebcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

No devices are held
```

- The `symmir establish` command initiates an incremental establish on the BCV pair, which refreshes the BCV with any new writes that occurred on DEV001 while the BCV pair was split.

```
symmir -g clonebcv establish DEV001 bcv ld CLONESRC -noprompt

'Incremental Establish' operation execution is in progress for device 'DEV001'
paired with BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...

'Incremental Establish' operation successfully initiated for device 'DEV001' in
group 'clonebcv' paired with BCV device 'CLONESRC'.
```

- The `symmir verify` command verifies that the establish operation is complete.

```
symmir -g clonebcv verify DEV001

Device 'DEV001' in group 'clonebcv' is in the 'Synchronized or Restored' state.
```

- The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the next clone operation.

```
symmir -g clonebcv split DEV001 -instant -not_ready -noprompt

'Split' operation execution is in progress for device 'DEV001' in device group
'clonebcv'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'clonebcv'.
```

- The `symmir verify` command checks every five seconds and verifies when the background split completes.

```
symmir -g clonebcv verify DEV001 -split -bg -i 5

Device 'DEV001' in group 'clonebcv' has finished splitting in the background.
```

- The following command creates a new CopyOnAccess copy session for source device CLONESRC and target device CLONETGT, in preparation for cloning the refreshed data on the BCV source device.

```
symclone -g clonebcv create CLONESRC sym ld CLONETGT -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

- The `symclone activate` command activates the clone operation for the clone pair.

```
symclone -g clonebcv activate CLONESRC sym ld CLONETGT -noprompt

'Activate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Activate' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

- The following command makes the BCV device ready again to its host. Reads or writes to the BCV source will clone data to the target.

```
symld -g clonebcv ready -bcv CLONESRC -noprompt

'Ready' Device operation successfully completed for the device.
```

- After cloning operations are complete, terminate the clone pair session.

```
symclone -g clonebcv terminate CLONESRC sym ld CLONETGT -noprompt -v

'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT' in device group 'clonebcv'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0215(S) - 0102(T) [PAIRED]

STARTING a Clone 'TERMINATE' operation.

The Clone 'TERMINATE' operation SUCCEEDED.

'Terminate' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

- The following commands perform cleanup tasks so that these same devices are available for use in "Cloning multiple copies of a BCV" on page 257, which follows. The `symmir cancel` command cancels the BCV pair relationship between DEV001 and CLONESRC so that the Symmetrix array no longer recognizes this relationship.

```
symmir -g clonebcv cancel DEV001 bcv ld CLONESRC

Execute 'Cancel' operation for device 'DEV001' in device group 'clonebcv'
(y/[n]) ? y

'Cancel' operation execution is in progress for device 'DEV001' paired with
BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...

'Cancel' operation successfully executed for device 'DEV001' in group
'clonebcv' paired with BCV device 'CLONESRC'.
```

- The `symdg delete` command deletes the device group `clonebcv`. The `-force` option is required when there are still devices in the device group.

  symdg -force delete clonebcv

Example 4: Cloning multiple copies of a BCV

This example creates a new device group and pairs Symmetrix standard device 00FF with BCV 0215 as a BCV pair. The clone source is BCV 0215. The targets are standard devices 0100, 0101, 0102, and 0103.

- The `symdg` command creates a device group named multiclone. The first `symld` command adds standard device 00FF to the group. By default, the logical device name of device 00FF is DEV001. The `symbcv` command associates the source BCV with the group. The example assigns CLONESRC as the BCV's logical device name to make it easier to identify the clone source device. The remaining `symld` commands add four target standard devices (0100, 0101, 0102, and 0103) and assign CLONETGT logical device names that make it easier to identify the clone target devices.

```
symdg create multiclone
symld -g multiclone add dev FF
symbcv -g multiclone associate dev 215 CLONESRC
symld -g multiclone add dev 100 CLONETGT1
symld -g multiclone add dev 101 CLONETGT2
symld -g multiclone add dev 102 CLONETGT3
symld -g multiclone add dev 103 CLONETGT4
```

- The `symdg show` command displays the details of the device group named multiclone.

```
symdg show multiclone

Group Name:  multiclone

    Group Type                                     : REGULAR
    Valid                                          : Yes
    Symmetrix ID                                   : 000187900041
    Group Creation Time                            : Tue Mar 18 14:51:12 2003
    Vendor ID                                      : EMC Corp
    Application ID                                 : SYMCLI

    Number of STD Devices in Group                 :    5
    Number of Associated GK's                      :    0
    Number of Locally-associated BCV's             :    1
    Number of Locally-associated VDEV's            :    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0

    Standard (STD) Devices (5):
        {
        --------------------------------------------------------------------
                                                        Sym                Cap
        LdevName              PdevName                  Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/vx/rdmp/c4t0d158s2 00FF       RW       4315
        CLONETGT1             /dev/vx/rdmp/c4t0d159s2 0100       RW       4315
        CLONETGT2             /dev/vx/rdmp/c4t0d160s2 0101       RW       4315
        CLONETGT3             /dev/vx/rdmp/c4t0d161s2 0102       RW       4315
        CLONETGT4             /dev/vx/rdmp/c4t0d162s2 0103       RW       4315
        }
    BCV Devices Locally-associated (1):
        {
        --------------------------------------------------------------------
                                                        Sym                Cap
        LdevName              PdevName                  Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        CLONESRC              N/A                       0215       RW       4315
        }
```

- The `symmir establish` command initiates a full establish operation on the BCV pair identified by their logical device names (standard device DEV001 with the BCV named CLONESRC).

```
symmir -g multiclone establish -full DEV001 bcv ld CLONESRC -noprompt

'Full Establish' operation execution is in progress for device 'DEV001' paired with
BCV device 'CLONESRC' in device group 'multiclone'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in group
'multiclone' paired with BCV device 'CLONESRC'.
```

- The `symmir verify` command checks the establish operation every 30 seconds and verifies when the operation is complete. That is, DEV001 has been fully copied to CLONESRC.

```
symmir -g multiclone verify DEV001 -i 30

Device 'DEV001' in group 'multiclone' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'multiclone' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'multiclone' is in the 'Synchronized or Restored' state.
```

- The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation.

```
symmir -g multiclone split -instant -not_ready DEV001 bcv ld CLONESRC -noprompt

'Split' operation execution is in progress for device 'DEV001' paired with BCV
device 'CLONESRC' in device group 'multiclone'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'multiclone'
paired with BCV device 'CLONESRC'.
```

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

```
symmir -g multiclone verify -bg -split DEV001 -i 5

Device 'DEV001' in group 'multiclone' has finished splitting in the background.
```

- The following command creates a copy session for source device CLONESRC and target device CLONETGT1, specifying the `-copy` option for a full copy of the source device.

```
symclone -g multiclone create -copy CLONESRC sym ld CLONETGT1 -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT1' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT1'.
```

- The following command creates a copy session for source device CLONESRC and target device CLONETGT2, specifying the `-copy` option for a full copy of the source device.

```
symclone -g multiclone create -copy CLONESRC sym ld CLONETGT2 -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT2' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT2'.
```

- The following command creates a copy session for source device CLONESRC and target device CLONETGT3, specifying the `-copy` option for a full copy of the source device.

```
symclone -g multiclone create -copy CLONESRC sym ld CLONETGT3 -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT3' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT3'.
```

- The following command creates a copy session for source device CLONESRC and target device CLONETGT4, specifying the `-copy` option for a full copy of the source device.

```
symclone -g multiclone create -copy CLONESRC sym ld CLONETGT4 -noprompt

'Create' operation execution is in progress for device 'CLONESRC' paired with target
device 'CLONETGT4' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT4'.
```

- The `symclone query` command with the `-multi` option displays that the source device and its four clone targets have been successfully created.

  `symclone -g multiclone query -multi`

```
Device Group (DG) Name: multiclone
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device            Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical   Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

CLONESRC  0215     138090 CLONETGT4    0103 XX Created         0
                   138090 CLONETGT3    0102 XX Created         0
                   138090 CLONETGT2    0101 XX Created         0
                   138090 CLONETGT1    0100 XX Created         0

Total            --------
  Track(s)         552360
  MB(s)           17261.2

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The following `symclone activate` command activates the four copy sessions at the same time.

```
symclone -g multiclone activate CLONESRC sym ld CLONETGT1
                                CLONESRC sym ld CLONETGT2
                                CLONESRC sym ld CLONETGT3
                                CLONESRC sym ld CLONETGT4

Execute 'Activate' operation for the 4 specified devices in device
group
'multiclone' (y/[n]) ? y

'Activate' operation execution is in progress for the device list in
device group 'multiclone'. Please wait...

'Activate' operation successfully executed for the device list in
device group 'multiclone'.
```

- The following command makes the BCV device ready again to its host.

  `symld -g multiclone ready -bcv CLONESRC -noprompt`

```
'Ready' Device operation successfully completed for the device.
```

- The following `symclone query` command with the `-multi` option displays the state of the clone operation, showing that all four copy operations are still in progress (CopyInProg) and the completion percent of each.

  `symclone -g multiclone query -multi`

```
Device Group (DG) Name: multiclone
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900041

      Source Device            Target Device        State     Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym   Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

CLONESRC   0215      129275 CLONETGT1   0100 XX CopyInProg     6
                     129277 CLONETGT2   0101 XX CopyInProg     6
                     129276 CLONETGT3   0102 XX CopyInProg     6
                     129276 CLONETGT4   0103 XX CopyInProg     6

Total              --------
  Track(s)           517104
  MB(s)             16159.5

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The `symclone verify` command checks all clone pairs every 60 seconds until all are in the Copied state.

  `symclone -g multiclone verify CLONESRC -copied -i 60`

```
Device 'CLONESRC' in group 'multiclone' is NOT in the 'Copied' state.

Device 'CLONESRC' in group 'multiclone' is NOT in the 'Copied' state.

Device 'CLONESRC' in group 'multiclone' is in the 'Copied' state.
```

- The following commands terminate the copy sessions. Stopping a copy session deletes clone pair information from the SYMAPI database and removes any hold on a target device.

```
symclone -g multiclone terminate CLONESRC sym ld CLONETGT1 -noprompt

'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT1' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT1'.

symclone -g multiclone terminate CLONESRC sym ld CLONETGT2 -noprompt

'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT2' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT2'.

symclone -g multiclone terminate CLONESRC sym ld CLONETGT3 -noprompt

'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT3' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT3'.

symclone -g multiclone terminate CLONESRC sym ld CLONETGT4 -noprompt

'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT4' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT4'.
```

- The `symmir establish` command initiates an incremental establish on the BCV pair, which refreshes the BCV with any new writes that occurred on DEV001 while the BCV pair was split.

```
symmir -g multiclone establish DEV001 bcv ld CLONESRC -noprompt

'Incremental Establish' operation execution is in progress for
device 'DEV001' paired with BCV device 'CLONESRC' in device group
'multiclone'. Please wait...

'Incremental Establish' operation successfully initiated for device
'DEV001' in group 'multiclone' paired with BCV device 'CLONESRC'.
```

- The `symmir verify` command checks the status of the establish operation every 30 seconds and verifies when the operation is complete.

```
symmir -g multiclone verify DEV001 -i 30

Device 'DEV001' in group 'multiclone' is in the 'Synchronized or
Restored' state.
```

Example 5: Cloning 16 copies from a single source device

This example creates a new device group containing a single standard device (023E) as a clone source and 16 BCV devices (0257 – 0266) as clone targets.

- The `symdg` command creates a device group named clone16. The `symld` command adds standard device 023E to the group. By default, the logical device name of device 023E is DEV001. The `symbcv` command associates a range of 16 BCV devices (0257 through 0266) with the group and assigns them the logical device names BCV001 through BCV016.

```
symdg create clone16
symld -g clone16 add dev 23e
symbcv -g clone16 associateall dev -range 257:266
```

- When cloning to multiple targets from the same source device, you must create each of the 16 source/target pairings separately. The first `symclone create` command creates the DEV001/BCV001 clone pair and displays the resulting output from the command. The rest of the `symclone create` commands create the remaining 15 clone pairs (these outputs are omitted for brevity). By default, each clone pair session created will be a CopyOnAccess session.

```
symclone -g clone16 create DEV001 sym ld BCV001 -noprompt

'Create' operation execution is in progress for device 'DEV001'
paired with
target device 'BCV001' in device group 'clone16'. Please wait...

'Create' operation successfully executed for device 'DEV001' in
group
'clone16' paired with target device 'BCV001'.

symclone -g clone16 create DEV001 sym ld BCV002 -noprompt
symclone -g clone16 create DEV001 sym ld BCV003 -noprompt
symclone -g clone16 create DEV001 sym ld BCV004 -noprompt
symclone -g clone16 create DEV001 sym ld BCV005 -noprompt
symclone -g clone16 create DEV001 sym ld BCV006 -noprompt
symclone -g clone16 create DEV001 sym ld BCV007 -noprompt
symclone -g clone16 create DEV001 sym ld BCV008 -noprompt
symclone -g clone16 create DEV001 sym ld BCV009 -noprompt
symclone -g clone16 create DEV001 sym ld BCV010 -noprompt
symclone -g clone16 create DEV001 sym ld BCV011 -noprompt
symclone -g clone16 create DEV001 sym ld BCV012 -noprompt
symclone -g clone16 create DEV001 sym ld BCV013 -noprompt
symclone -g clone16 create DEV001 sym ld BCV014 -noprompt
symclone -g clone16 create DEV001 sym ld BCV015 -noprompt
symclone -g clone16 create DEV001 sym ld BCV016 -noprompt
```

- The `symclone query` command with the `-multi` option displays that the source device and its 16 clone targets have been successfully created. Note that the last clone pair created is located at the top of the display list.

symclone -g clone16 query -multi

```
Device Group (DG) Name: clone16
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000005231

      Source Device            Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical    Sym  Tracks    Logical      Sym  CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001    023E     138090 BCV016      0266 XX Created         0
                   138090 BCV015      0265 XX Created         0
                   138090 BCV014      0264 XX Created         0
                   138090 BCV013      0263 XX Created         0
                   138090 BCV012      0262 XX Created         0
                   138090 BCV011      0261 XX Created         0
                   138090 BCV010      0260 XX Created         0
                   138090 BCV009      025F XX Created         0
                   138090 BCV008      025E XX Created         0
                   138090 BCV007      025D XX Created         0
                   138090 BCV006      025C XX Created         0
                   138090 BCV005      025B XX Created         0
                   138090 BCV004      025A XX Created         0
                   138090 BCV003      0259 XX Created         0
                   138090 BCV002      0258 XX Created         0
                   138090 BCV001      0257 XX Created         0

Total             --------
  Track(s)         2209440
  MB(s)            69045.0

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- The following `symclone activate` command activates the 16 clone pair sessions at the same time. When you include the `-consistent` option, SYMAPI suspends any further I/O before activation. Suspending I/O during the brief period when activation occurs prevents any pending records from being included in the clone copy. When activation completes, SYMAPI allows I/O to resume.

```
symclone -g clone16 activate -consistent
                   DEV001 sym ld BCV001
                   DEV001 sym ld BCV002
                   DEV001 sym ld BCV003
                   DEV001 sym ld BCV004
                   DEV001 sym ld BCV005
                   DEV001 sym ld BCV006
                   DEV001 sym ld BCV007
                   DEV001 sym ld BCV008
                   DEV001 sym ld BCV009
                   DEV001 sym ld BCV010
                   DEV001 sym ld BCV011
                   DEV001 sym ld BCV012
                   DEV001 sym ld BCV013
                   DEV001 sym ld BCV014
                   DEV001 sym ld BCV015
                   DEV001 sym ld BCV016 -noprompt

'Activate' operation execution is in progress for the device list in
device group 'clone16'. Please wait...

'Activate' operation successfully executed for the device list in
device group 'clone16'.
```

- The `symclone query` command with the `-multi` option displays that the source device and its 16 clone targets are in the CopyOnAccess state. You can now clone data to any four targets simultaneously. You can continue to clone data to these targets as long as these copy sessions exist.

`symclone -g clone16 query -multi`

```
Device Group (DG) Name: clone16
DG's Type           : REGULAR
DG's Symmetrix ID   : 000000005231

      Source Device           Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Logical     Sym   Tracks    Logical     Sym   CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----

DEV001     023E      138090 BCV016      0266 XX CopyOnAccess   0
                     138090 BCV015      0265 XX CopyOnAccess   0
                     138090 BCV014      0264 XX CopyOnAccess   0
                     138090 BCV013      0263 XX CopyOnAccess   0
                     138090 BCV012      0262 XX CopyOnAccess   0
                     138090 BCV011      0261 XX CopyOnAccess   0
                     138090 BCV010      0260 XX CopyOnAccess   0
                     138090 BCV009      025F XX CopyOnAccess   0
                     138090 BCV008      025E XX CopyOnAccess   0
                     138090 BCV007      025D XX CopyOnAccess   0
                     138090 BCV006      025C XX CopyOnAccess   0
                     138090 BCV005      025B XX CopyOnAccess   0
                     138090 BCV004      025A XX CopyOnAccess   0
                     138090 BCV003      0259 XX CopyOnAccess   0
                     138090 BCV002      0258 XX CopyOnAccess   0
                     138090 BCV001      0257 XX CopyOnAccess   0

Total                --------
  Track(s)            2209440
  MB(s)               69045.0

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.
```

- When you are finished cloning data to these targets, you need to terminate each of the 16 source/target pairings separately. The first `symclone terminate` command terminates the DEV001/BCV001 clone pair session and displays the resulting output from the command. The rest of the `symclone terminate` commands terminate the remaining 15 clone pairs (these outputs are omitted for brevity).

`symclone -g clone16 terminate DEV001 sym ld BCV001 -noprompt`

```
'Terminate' operation execution is in progress for device 'DEV001' paired with
target device 'BCV001' in device group 'clone16'. Please wait...

'Terminate' operation successfully executed for device 'DEV001' in group
'clone16' paired with target device 'BCV001'.
```

`symclone -g clone16 terminate DEV001 sym ld BCV002 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV003 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV004 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV005 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV006 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV007 -noprompt`
`symclone -g clone16 terminate DEV001 sym ld BCV008 -noprompt`

```
symclone -g clone16 terminate DEV001 sym ld BCV009 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV010 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV011 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV012 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV013 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV014 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV015 -noprompt
symclone -g clone16 terminate DEV001 sym ld BCV016 -noprompt
```

Example 6: Using a composite group to control clone pairs

This example was performed using Solutions Enabler Version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (sid 35 and sid 60). A composite group defined on the host contains a set of clone pairs that spans the two Symmetrix arrays.

You can create, activate, and control specific clone pairs within the composite group as well as performing control operations on the entire group as shown here. For example, you can terminate one clone session in the group without terminating the group's other sessions.)

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

  ```
  symcg create TFClone -type regular
  ```

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

  ```
  symcg -cg TFClone addall dev -range 56:6E -sid 35
  symcg -cg TFClone addall dev -range 14:27 -sid 60
  ```

- The following `symbcv` commands add to the composite group a range of BCV devices from each of the two source Symmetrix arrays. These are the target devices.

  ```
  symbcv -cg TFClone associateall dev -range 182:19A -sid 35
  symbcv -cg TFClone associateall dev -range 3B6:3C9 -sid 60
  ```

- The `symclone create` command creates optimized standard/BCV pairings among devices within each Symmetrix array and places each clone pair in the Created state. The `-copy` option specifies that full copying be performed upon activation.

  ```
  symclone -cg TFClone create -copy -optimize -noprompt

  'Create' operation execution is in progress for composite group
  'TFClone'.
  Please wait...

  'Create' operation successfully executed for composite group
  'TFClone'.
  ```

- The `symclone query` command displays the clone pairs and that they are in the Created state. Note that devices 0056 and 0059 are meta devices (as are their target devices), and only the meta head device is displayed.

symclone -cg TFClone query

```
Composite Group Name       :  TFClone
Composite Group Type       :  REGULAR
Number of Symmetrix Units  :  2
Number of RDF (RA) Groups  :  2

Symmetrix ID            : 000187900035    (Microcode Version: 5670)

      Source Device          Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Sym             Tracks    Sym               CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----
0056               414270 0182              XX Created        0
0059               276180 0185              XX Created        0
005B               138090 0187              XX Created        0
005C               138090 0188              XX Created        0
005D               138090 0189              XX Created        0
005E               138090 018A              XX Created        0
005F               138090 018B              XX Created        0
0060               138090 018C              XX Created        0
0061               138090 018D              XX Created        0
0062               138090 018E              XX Created        0
0063               138090 018F              XX Created        0
0064               138090 0190              XX Created        0
0065               138090 0191              XX Created        0
0066               138090 0192              XX Created        0
0067               138090 0193              XX Created        0
0068               138090 0194              XX Created        0
0069               138090 0195              XX Created        0
006A               138090 0196              XX Created        0
006B               138090 0197              XX Created        0
006C               138090 0198              XX Created        0
006D               138090 0199              XX Created        0
006E               138090 019A              XX Created        0

Symmetrix ID            : 000184500160    (Microcode Version: 5568)

      Source Device          Target Device        State      Copy
------------------------- ------------------- ------------ ----
                Protected
Sym             Tracks    Sym               CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----
0014               138090 03B6              XX Created        0
0015               138090 03B7              XX Created        0
0016               138090 03B8              XX Created        0
0017               138090 03B9              XX Created        0
0018               138090 03BA              XX Created        0
0019               138090 03BB              XX Created        0
001A               138090 03BC              XX Created        0
001B               138090 03BD              XX Created        0
001C               138090 03BF              XX Created        0
001D               138090 03BE              XX Created        0
001E               138090 03C0              XX Created        0
001F               138090 03C1              XX Created        0
0020               138090 03C2              XX Created        0
0021               138090 03C3              XX Created        0
0022               138090 03C4              XX Created        0
0023               138090 03C5              XX Created        0
0024               138090 03C6              XX Created        0
0025               138090 03C7              XX Created        0
```

```
0026                138090 03C8             XX Created         0
0027                138090 03C9             XX Created         0

Total              --------
  Track(s)         6214050
  MB(s)             194189

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with a composite group.
```

- The `symclone activate` command places each clone pair in the CopyInProg state. Full copying of the data begins immediately. The `-consistent` option causes the clone pairs to be consistently activated.

symclone -cg TFClone activate -consistent -noprompt

```
'Activate' operation execution is in progress for composite group 'TFClone'. Please
wait...

'Activate' operation successfully executed for composite group 'TFClone'.
```

- The `symclone query` command displays the clone pairs and their CopyInProg state.

symclone -cg TFClone query

```
Composite Group Name      :  TFClone
Composite Group Type      :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID            : 000187900035    (Microcode Version: 5670)

      Source Device           Target Device        State      Copy
------------------------- ------------------- ------------ ----
                 Protected
Sym              Tracks     Sym               CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----
0056                412378 0182             XX CopyInProg     0
0059                276180 0185             XX CopyInProg     0
005B                138090 0187             XX CopyInProg     0
005C                138090 0188             XX CopyInProg     0
005D                135506 0189             XX CopyInProg     1
005E                138090 018A             XX CopyInProg     0
005F                136329 018B             XX CopyInProg     1
0060                138090 018C             XX CopyInProg     0
0061                138090 018D             XX CopyInProg     0
0062                138090 018E             XX CopyInProg     0
0063                137681 018F             XX CopyInProg     0
0064                138090 0190             XX CopyInProg     0
0065                138090 0191             XX CopyInProg     0
0066                136981 0192             XX CopyInProg     0
0067                138090 0193             XX CopyInProg     0
0068                138090 0194             XX CopyInProg     0
0069                137020 0195             XX CopyInProg     0
006A                138090 0196             XX CopyInProg     0
006B                138090 0197             XX CopyInProg     0
006C                138090 0198             XX CopyInProg     0
006D                138090 0199             XX CopyInProg     0
006E                138090 019A             XX CopyInProg     0

Symmetrix ID            : 000184500160    (Microcode Version: 5568)

      Source Device           Target Device        State      Copy
```

```
------------------------- ------------------- ------------ ----
                Protected
Sym             Tracks    Sym                CG SRC <=> TGT  (%)
------------------------- ------------------- ------------ ----
0014               136681 03B6               XX CopyInProg     1
0015               138090 03B7               XX CopyInProg     0
0016               138090 03B8               XX CopyInProg     0
0017               138090 03B9               XX CopyInProg     0
0018               138090 03BA               XX CopyInProg     0
0019               138090 03BB               XX CopyInProg     0
001A               138090 03BC               XX CopyInProg     0
001B               138090 03BD               XX CopyInProg     0
001C               138090 03BF               XX CopyInProg     0
001D               138090 03BE               XX CopyInProg     0
001E               138090 03C0               XX CopyInProg     0
001F               138090 03C1               XX CopyInProg     0
0020               138090 03C2               XX CopyInProg     0
0021               137328 03C3               XX CopyInProg     0
0022               138090 03C4               XX CopyInProg     0
0023               138090 03C5               XX CopyInProg     0
0024               138090 03C6               XX CopyInProg     0
0025               138090 03C7               XX CopyInProg     0
0026               138090 03C8               XX CopyInProg     0
0027               138090 03C9               XX CopyInProg     0

Total             --------
  Track(s)         6203054
  MB(s)             193845

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with a composite group.
```

- The `symclone terminate` command attempts to end all clone sessions in the composite group but cannot do so because the clone pairs are in the CopyInProg state.

```
symclone -cg TFClone terminate -noprompt

'Terminate' operation execution is in progress for composite group 'TFClone'. Please
wait...

The session cannot be terminated because the device is 'Copy In Progress' or
'Restore In Progress' unless the Symmetrix force flag is used
```

- The `symclone terminate` command with the `-symforce` option forces all clone sessions in the composite group to terminate.

```
symclone -cg TFClone terminate -symforce -noprompt

'Terminate' operation execution is in progress for composite group 'TFClone'. Please
wait...

'Terminate' operation successfully executed for composite group 'TFClone'.
```

Example 7: Cloning to a remote BCV

This example was performed using Solutions Enabler Version 6.0. The hardware setup includes a host connected to a source Symmetrix array (sid 58) that is RDF-connected to a remote Symmetrix array (sid 35). A device group is defined on the host. The device group contains a single standard device (18D) as the clone source and a remote BCV device (2BB) as the clone target.

- The `symdg create` command creates an RDF1 type device group named Rdf1Grp. The `symld add` command adds to the device group an R1 standard device (18D) on the local Symmetrix (sid 58) to be the source device. The `symbcv associate` command associates a target BCV device (2BB) on the remote Symmetrix to hold the clone copy. A *remote* target device must be a BCV.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp add dev 18D -sid 58
symbcv -g Rdf1Grp associate dev 2BB -rdf
```

- The `symclone establish` command initiates an immediate full copy from the source standard device to the remote BCV target device via the remote R2 device (047B). The `-rdf` parameter is required to specify a remote target.

```
symclone -g Rdf1Grp establish -full -rdf -noprompt

Remote 'Clone Full Establish' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...

Remote 'Clone Full Establish' operation successfully executed for
device group
'Rdf1Grp'.
```

- The `symclone query` command with the `-rdf` parameter displays the progress of the remote clone operation. Note that the data is being cloned from the R1 source device via its paired R2 device (047B).

```
symclone -g Rdf1Grp query -rdf

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758
Remote Symmetrix ID   : 000187900035


          Source Device                       Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical    Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001     047B    118824        0 RBCV001    02BB        0 XXX. CopyInProg    13

Total            --------
  MB(s)            3713.2


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- The `symclone verify` command checks at 30-second intervals (`-i 30`) to determine when the clone pair reaches the Copied state. The ellipsis (...) indicates omitted output.

```
symclone -g Rdf1Grp verify -rdf -i 30

None of the devices in group 'Rdf1Grp' are in the 'Copied' state.

None of the devices in group 'Rdf1Grp' are in the 'Copied' state.

.....................................................................................

All devices in group 'Rdf1Grp' have a paired TGT device that is in
the 'Copied' state.
```

- A subsequent `symclone query` command confirms that the remote clone is in the Copied state.

```
symclone -g Rdf1Grp query -rdf

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758
Remote Symmetrix ID   : 000187900035


          Source Device                     Target Device          State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified              Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001    047B         0        0 RBCV001   02BB        0 XXX. Copied       100

Total             --------
  MB(s)               0.0


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- The `symclone terminate` command ends the clone copy session. Terminating a copy session removes any hold on the target device and deletes pair information about the terminated pair from the Symmetrix array.

  ```
  symclone -g Rdf1Grp terminate -rdf -noprompt

  Remote 'Terminate' operation execution is in progress for
  device group 'Rdf1Grp'. Please wait...

  Remote 'Terminate' operation successfully executed for device group
  'Rdf1Grp'.
  ```

- The `symdg delete` command deletes the device group so that this same group name can be used in subsequent examples. The `-force` option is required when the group is not empty.

  ```
  symdg delete Rdf1Grp -force
  ```

Example 8: Cloning to a local BCV and a remote BCV

This example was performed using Solutions Enabler Version 6.0. The hardware setup includes a host connected to a source Symmetrix array (sid 58) that is RDF-connected to a remote Symmetrix array (sid 35). A device group is defined on the host. The device group contains a single standard device (18D) as the clone source, and a local BCV (AB) and a remote BCV (2BB) as the clone targets.

- The `symdg create` command creates an RDF1 type device group named Rdf1Grp. The `symld add` command adds to the device group an R1 standard device (18D) on the local Symmetrix (sid 58) as the source device. The `symbcv associate` commands associate a local target BCV (AB) and a remote target BCV device (2BB) to hold clone copies. A *remote* target device must be a BCV.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp add dev 18D -sid 58
symbcv -g Rdf1Grp associate dev AB
symbcv -g Rdf1Grp associate dev 2BB -rdf
```

- The `symclone establish` command clones a full copy from the source standard device (18D) to the local BCV target device (AB). When the source and target devices of a device group are not specified on the command line, the assumption is that the source is a standard device, and the target is a BCV.

```
symclone -g Rdf1Grp establish -full -noprompt

'Clone Full Establish' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...

'Clone Full Establish' operation successfully executed for device
group
'Rdf1Grp'.
```

- The `symclone query` command shows that the local clone copy operation is in progress.

```
symclone -g Rdf1Grp query

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758


          Source Device                       Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks  Logical    Sym  tracks    CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001     018D     87276        0 BCV001     00AB        0 XXX. CopyInProg    36

Total            --------
  MB(s)            2727.4

Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- An SRDF `symrdf query` command shows that the SRDF pair (the local R1 standard device and its remote R2 device) are synchronized.

```
symrdf -g Rdf1Grp query

Device Group (DG) Name              : Rdf1Grp
DG's Type                           : RDF1
DG's Symmetrix ID                   : 000187720758


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

DEV001  018D RW       0        0 RW 047B NR       0        0 S..   Synchronized

Total          -------- --------           -------- --------
  MB(s)             0.0      0.0                0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

- The `symclone establish` command initiates an immediate full copy from the source device to the remote BCV target via the remote R2 device (047B). The `-rdf` parameter is required to specify a remote target.

```
symclone -g Rdf1Grp establish -full -rdf -noprompt

Remote 'Clone Full Establish' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...

Remote 'Clone Full Establish' operation successfully executed for device group
'Rdf1Grp'.
```

- The `symclone query` command shows that the remote clone copy operation is in progress.

```
symclone -g Rdf1Grp query -rdf

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758
Remote Symmetrix ID   : 000187900035

          Source Device                     Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical    Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001    047B     69735         0 RBCV001   02BB         0 XXX. CopyInProg   49

Total           --------
  MB(s)           2179.2

Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- Once the local and remote clone operations are complete, you can terminate the two clone sessions. The following command terminates the local clone session.

```
symclone -g Rdf1Grp terminate -noprompt

'Terminate' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...

'Terminate' operation successfully executed for device group
'Rdf1Grp'.
```

- The following command uses the `-rdf` parameter to terminate the remote clone session.

```
symclone -g Rdf1Grp terminate -rdf -noprompt

Remote 'Terminate' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...

Remote 'Terminate' operation successfully executed for device group
'Rdf1Grp'.
```

- The `symdg delete` command deletes the device group as was done in Example 7 at completion.

```
symdg delete Rdf1Grp -force
```

Example 9: Cloning to multiple local and remote BCVs

This example was performed using Solutions Enabler Version 6.0. The hardware setup includes a host connected to a source Symmetrix array (sid 58) that is RDF-connected to a remote Symmetrix array (sid 35). A device group is defined on the host. The device group contains a single standard device (18D) as the clone source and a range of four local BCVs and four remote BCVs as the clone targets.

- The `symdg create` command creates an RDF1 type device group named Rdf1Grp. The `symld add` command adds to the device group an R1 standard device (18D) on the local Symmetrix (sid 58) to be the source device. The `symbcv associateall` commands associate a range of four local target BCVs (A8 through AB) and four remote target BCVs (8D through 90) to hold the clone copies. A *remote* target device must be a BCV.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp add dev 18D -sid 58
symbcv -g Rdf1Grp associateall dev -range A8:AB
symbcv -g Rdf1Grp associateall dev -range 8D:90 -rdf
```

- To clone the four *local* target devices from the local source device, first issue four `symclone create` commands, specifying the same source device with each of the four targets. The following command creates a clone session for the first local clone pair (DEV001 and BCV001). Setting the `-precopy` mode of operation initiates data copying in the background prior to activating the clone session, allowing the early movement of data prior to completing the full clone copy. The `-diff` option (`-differential`) is included here in case a subsequent incremental clone operation needs to be performed.

```
symclone -g Rdf1Grp create -precopy -diff -noprompt DEV001 sym ld
BCV001

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'BCV001' in
device group 'Rdf1Grp'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV001'.
```

- The following command creates a clone session for the second local clone pair (DEV001 and BCV002).

```
symclone -g Rdf1Grp create -precopy -diff -noprompt DEV001 sym ld
BCV002

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'BCV002' in
device group 'Rdf1Grp'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV002'.
```

- The following command creates a clone session for the third local clone pair (DEV001 and BCV003).

```
symclone -g Rdf1Grp create -precopy -diff -noprompt DEV001 sym ld
BCV003
'Create' operation execution is in progress for device 'DEV001'
paired with target device 'BCV003' in
device group 'Rdf1Grp'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV003'.
```

- The following command creates a clone session for the fourth local clone pair (DEV001 and BCV004).

```
symclone -g Rdf1Grp create -precopy -diff -noprompt DEV001 sym ld BCV004

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'BCV004' in
device group 'Rdf1Grp'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV004'.
```

- The `symclone query` command with the `-multi` option shows the status of the four local clone sessions. Each clone pair is in the PreCopy state and has various percents of copy completion.

```
symclone -g Rdf1Grp query -multi

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758


          Source Device                        Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical    Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001     018D    101661         0 BCV004     00AB        0 XXX. PreCopy       26
                    64684         0 BCV003     00AA        0 XXX. PreCopy       53
                    65909         0 BCV002     00A9        0 XXX. PreCopy       52
                    54453         0 BCV001     00A8        0 XXX. PreCopy       60

Total               --------
  Track(s)           286707
  MB(s)              8959.6


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- You can activate the clone operation with one command that activates all sessions at the same time. Including the `-consistent` option means that SYMAPI will suspend I/O before activation, preventing any pending records from being included in the copy. After activation occurs, SYMAPI allows I/O to resume. The target devices will contain a copy of the source device that is consistent with the production database up until the time of the activation. The production DBMS instance has no knowledge that the suspend/ resume operation occurred.

```
symclone -g Rdf1Grp activate -noprompt -consistent DEV001 sym ld BCV001
                                                   DEV001 sym ld BCV002
                                                   DEV001 sym ld BCV003
                                                   DEV001 sym ld BCV004

'Activate' operation execution is in progress for the device list in
device group 'Rdf1Grp'. Please wait...

'Activate' operation successfully executed for the device list
in device group 'Rdf1Grp'.
```

- The `symclone verify` command checks if the clone pairs in the group are all in the Copied state.

```
symclone -g Rdf1Grp verify DEV001

Device 'DEV001' in group 'Rdf1Grp' is in the 'Copied' state(s).
```

- The `symclone query` command confirms that the local clone copy operations are complete.

```
symclone -g Rdf1Grp query -multi

Device Group (DG) Name: Rdf1Grp
DG's Type           : RDF1
DG's Symmetrix ID   : 000187720758


          Source Device                     Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
                Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical   Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001     018D         0        0 BCV001    00A8         0 XXX. Copied       100
                        0        0 BCV002    00A9         0 XXX. Copied       100
                        0        0 BCV003    00AA         0 XXX. Copied       100
                        0        0 BCV004    00AB         0 XXX. Copied       100

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


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- To clone the four *remote* target devices from the local source device, first issue four `symclone create` commands, specifying the same source device with each of the four remote targets. The following command creates a clone session for the first remote clone pair (DEV001 and RBCV001). Setting the `-precopy` mode of operation initiates data copying in the background prior to activating a clone session. The `-rdf` parameter is required to indicate that this is a remote clone pair.

```
symclone -g Rdf1Grp create -precopy -diff -rdf DEV001 sym ld RBCV001 -noprompt

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV001' in
device group 'Rdf1Grp'. Please wait...

Remote 'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV001'.
```

- The following command creates a clone session for the second remote clone pair (DEV001 and RBCV002)

```
symclone -g Rdf1Grp create -precopy -diff -rdf DEV001 sym ld RBCV002 -noprompt

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV002' in
device group 'Rdf1Grp'. Please wait...

Remote 'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV002'.
```

- The following command creates a clone session for the third remote clone pair (DEV001 and RBCV003)

```
symclone -g Rdf1Grp create -precopy -diff -rdf DEV001 sym ld RBCV003 -noprompt

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV003' in
device group 'Rdf1Grp'. Please wait...

Remote 'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV003'.
```

- The following command creates a clone session for the fourth remote clone pair (DEV001 and RBCV004)

```
symclone -g Rdf1Grp create -precopy -diff -rdf DEV001 sym ld RBCV004 -noprompt

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV004' in
device group 'Rdf1Grp'. Please wait...

Remote 'Create' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV004'.
```

- The `symclone query` command with the `-multi` and `-rdf` options shows the status of the four remote clone sessions. Although each clone pair is in the PreCopy state, each pair has already reached 100 percent copy completion.

```
symclone -g Rdf1Grp query -rdf -multi

Device Group (DG) Name: Rdf1Grp
DG's Type              : RDF1
DG's Symmetrix ID      : 000187720758
Remote Symmetrix ID    : 000187900035

          Source Device                        Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
                 Protected Modified                Modified
Logical    Sym  Tracks     Tracks   Logical    Sym  tracks    CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001     047B            0         0 RBCV004    0090          0 XXXX PreCopy      100
                           0         0 RBCV003    008F          0 XXXX PreCopy      100
                           0         0 RBCV002    008E          0 XXXX PreCopy      100
                           0         0 RBCV001    008D          0 XXXX PreCopy      100

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


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- The `symclone activate` command activates all sessions for the remote clone pairs simultaneously. Ordinarily, activation of a full clone copy initiates the completion of the clone operation that began in precopy mode. But the copying for each session is already 100 percent complete. In this case, the activation simply transitions the clone pair states from PreCopy to Copied.

```
symclone -g Rdf1Grp activate -noprompt -consistent -rdf DEV001 sym ld RBCV001
                                                        DEV001 sym ld RBCV002
                                                        DEV001 sym ld RBCV003
                                                        DEV001 sym ld RBCV004

Remote 'Activate' operation execution is in progress for the device list in
device group 'Rdf1Grp'. Please wait...

Remote 'Activate' operation successfully executed for the device list
in device group 'Rdf1Grp'.
```

- The `symclone verify` command checks if the clone pairs in the group are all in the Copied state.

```
symclone -g Rdf1Grp verify DEV001 -rdf

Device 'DEV001' in group 'Rdf1Grp' is in the 'Copied' state(s).
```

- The `symclone query` command confirms that the remote clone copy operations are complete.

```
symclone -g Rdf1Grp query -rdf -multi

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758
Remote Symmetrix ID   : 000187900035


          Source Device                     Target Device          State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

DEV001    047B         0        0 RBCV001   008D        0 XXX. Copied       100
                       0        0 RBCV002   008E        0 XXX. Copied       100
                       0        0 RBCV003   008F        0 XXX. Copied       100
                       0        0 RBCV004   0090        0 XXX. Copied       100

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


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- Once the local and remote clone operations are complete, you can terminate the clone sessions. The following four commands terminate the four local clone sessions.

```
symclone -g Rdf1Grp terminate -noprompt DEV001 bcv ld BCV001

'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'BCV001' in
device group 'Rdf1Grp'. Please wait...

'Terminate' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV001'.

symclone -g Rdf1Grp terminate -noprompt DEV001 bcv ld BCV002

'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'BCV002' in
device group 'Rdf1Grp'. Please wait...

'Terminate' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV002'.

symclone -g Rdf1Grp terminate -noprompt DEV001 bcv ld BCV003

'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'BCV003' in
device group 'Rdf1Grp'. Please wait...
```

```
'Terminate' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV003'.
```

symclone -g Rdf1Grp terminate -noprompt DEV001 bcv ld BCV004

```
'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'BCV004' in
device group 'Rdf1Grp'. Please wait...

'Terminate' operation successfully executed for device 'DEV001'
in group 'Rdf1Grp' paired with target device 'BCV004'.
```

- The following four commands with the `-rdf` parameter terminate the four remote clone sessions.

symclone -g Rdf1Grp terminate -noprompt -rdf DEV001 bcv ld RBCV001

```
'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV001' in
device group 'Rdf1Grp'. Please wait...

Remote 'Terminate' operation successfully executed for device
'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV001'.
```

symclone -g Rdf1Grp terminate -noprompt -rdf DEV001 bcv ld RBCV002

```
'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV002' in
device group 'Rdf1Grp'. Please wait...

Remote 'Terminate' operation successfully executed for device
'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV002'.
```

symclone -g Rdf1Grp terminate -noprompt -rdf DEV001 bcv ld RBCV003

```
'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV003' in
device group 'Rdf1Grp'. Please wait...

Remote 'Terminate' operation successfully executed for device
'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV003'.
```

symclone -g Rdf1Grp terminate -noprompt -rdf DEV001 bcv ld RBCV004

```
'Terminate' operation execution is in progress for device 'DEV001'
paired with target device 'RBCV004' in
device group 'Rdf1Grp'. Please wait...

Remote 'Terminate' operation successfully executed for device
'DEV001'
in group 'Rdf1Grp' paired with target device 'RBCV004'.
```

- The `symdg delete` command deletes the device group so that this same group name can be used in subsequent examples. The `-force` option is required when the group is not empty.

symdg delete Rdf1Grp -force

Example 10: Cloning to a remote BCV from a local BCV

This example was performed using Solutions Enabler Version 6.0. The hardware setup includes a host connected to a source Symmetrix array (sid 58) that is RDF-connected to a remote Symmetrix array (sid 35). A device group contains a single standard device (8B) paired with a local BCV (CC). This local BCV is the source of a clone operation to a remote BCV target (4C3). First, the example copies the data to the local BCV, then via SRDF to a remote R2 device, and then to the remote BCV clone.

- The `symdg create` command creates a Standard type device group named Group1. The `symld add` command adds to the device group a standard device (8B) on the local Symmetrix (sid 58) to be paired with a local BCV. The first `symbcv associate` command associates the local R1 BCV device (CC), which will be the source of the clone operation via its remote R2 device. Then the example associates the remote target BCV device (4C3) to hold the clone copy.

```
symdg create Group1
symld -g Group1 add dev 8B -sid 58
symbcv -g Group1 associate dev CC
symbcv -g Group1 associate dev 4C3 -bcv -rdf
```

- The `symclone establish` command clones a full copy from the source standard device (8B) to the local BCV target device (CC). When the source and target devices of a device group are not specified on the command line, the assumption is that the source is a standard device, and the target is a BCV.

```
symclone -g Group1 establish -full -noprompt

'Clone Full Establish' operation execution is in progress for
device group 'Group1'. Please wait...
```

- The `symclone verify` command checks at 30-second intervals (`-i 30`) to determine when the clone pair reaches the Copied state. The ellipsis (...) indicates omitted output.

```
symclone -g Group1 verify -i 30

None of the devices in group 'Group1' are in the 'Copied' state.
......................................................................
All devices in group 'Group1' have a paired TGT device that is in
the 'Copied' state.
```

- The SRDF `symrdf establish` command establishes the local R1 BCV with its remote R2 device. The `-bcv` parameter is required when the source device is an R1 BCV.

```
symrdf -g Group1 establish -bcv -noprompt

An RDF 'Incremental Establish' operation execution is
in progress for device group 'Group1'. Please wait...

    Suspend RDF link(s).......................................Done.
    Resume RDF
link(s)........................................Started.
    Merge device track tables between source and
target.......Started.
    Device: 00CC ............................................
Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s).......................................Done.
The RDF 'Incremental Establish' operation successfully initiated for
device group 'Group1'.
```

- The SRDF `symrdf query` command shows that the local R1 BCV and its remote R2 device are in process of becoming synchronized.

```
symrdf -g Group1 -bcv query

Device Group (DG) Name             : Group1
DG's Type                          : REGULAR
DG's Symmetrix ID                  : 000187720758
                     R E M O T E      S Y M M E T R I X


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

BCV001   00CC RW       0   136158 RW 03BA NR       0        0 S..   SyncInProg

Total           -------- --------          -------- --------
  MB(s)              0.0   4254.9               0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

- The SRDF `symrdf verify` command checks at 30-second intervals to determine when the R1 and R2 are fully Synchronized.

```
symrdf -g Group1 -bcv verify -i 30

NONE of the mirrored pairs are in the 'Synchronized' state.
........................................................................

All devices in the RDF group 'Group1' are in the 'Synchronized'
state.
```

- The `symclone establish` command clones a copy of the remote R2 device to the remote BCV.

```
symclone -g Group1 establish -full -bcv -rdf -noprompt

Remote 'Clone Full Establish' operation execution is in progress for
device group 'Group1'. Please wait...

Remote 'Clone Full Establish' operation successfully executed for
device group
'Group1'.
```

- The `symclone query` command shows that the remote clone copy operation from the R1 BCV source is actually being copied from the BCV's remote R2 device (3BA).

```
symclone -g Group1 query -bcv -rdf

Device Group (DG) Name: Group1
DG's Type              : REGULAR
DG's Symmetrix ID      : 000187720758
Remote Symmetrix ID    : 000187900035


          Source Device                        Target Device            State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical    Sym  tracks    CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

BCV001     03BA    121259        0 BRBCV001  04C3        0 XXX. CopyInProg    12

Total             --------
  MB(s)             3789.3


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- The `symclone verify` command checks at 30-second intervals (`-i 30`) to determine when the remote clone pair reaches the Copied state. The ellipsis (...) indicates omitted output.

```
symclone -g Group1 verify -bcv -rdf -i 30

None of the devices in group 'Group1' are in the 'Copied' state.
........................................................................................................
All devices in group 'Group1' have a paired TGT device that is in
the 'Copied' state.
```

- The `symclone terminate` command with the `-rdf` option terminates the remote clone session.

```
symclone -g Group1 terminate -bcv -rdf -noprompt

Remote 'Terminate' operation execution is in progress for
device group 'Group1'. Please wait...

Remote 'Terminate' operation successfully executed for device group
'Group1'.
```

- This command terminates the local clone session.

```
symclone -g Group1 terminate -noprompt

'Terminate' operation execution is in progress for
device group 'Group1'. Please wait...

'Terminate' operation successfully executed for device group
'Group1'.
```

- The `symdg delete` command deletes the device group as part of the normal clean-up process when you are finished using a particular device group. The `-force` option is required when the group is not empty.

  ```
  symdg delete Group1 -force
  ```

Example 11: Cloning in a multihop environment

This example was performed using Solutions Enabler Version 6.0 and a variety of TimeFinder/Mirror, TimeFinder/Clone, and SRDF commands. The hardware setup includes a host connected to a source Symmetrix array (sid 58) that is RDF-connected to a remote Symmetrix array (a.k.a. Hop 1). Hop 1 is RDF-connected to a third Symmetrix array (Hop 2). First, the example copies the data via SRDF to a remote R2 device on Hop 1, and then from that R2 to an R1 BCV there. Next, the example clones a copy from that R1 BCV's remote R2 device to a BCV on the Hop-2 Symmetrix array.

- The `symdg create` command creates an RDF1 type device group named Rdf1Grp. The `symld add` command adds to the device group an R1 standard device (18D) on the local Symmetrix (sid 58) to be the source device. The first `symbcv associate` command associates the R1 BCV (2BB) on the Hop-1 array, using the `-rdf` option to identify the BCV's location. The second `symbcv associate` command uses the `-rrdf` option to associate the Hop-2 BCV (22F) that will hold the clone copy.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp add dev 18D -sid 58
symbcv -g Rdf1Grp associate dev 2BB -rdf
symbcv -g Rdf1Grp associate dev 22F -rrdf
```

- The `symrdf query` command shows that the RDF link between the local source array and Hop 1 is functioning normally and that the source R1 device and its remote R2 are synchronized:

```
symrdf -g Rdf1Grp query

Device Group (DG) Name              : Rdf1Grp
DG's Type                           : RDF1
DG's Symmetrix ID                   : 000187720758


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

DEV001  018D RW       0        0 RW 047B WD       0        0 S..   Synchronized

Total           -------- --------           -------- --------
  MB(s)              0.0      0.0                0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

- The `symmir establish` command with the `-rdf` option fully establishes the remote R2 device on Hop 1 with the BCV there.

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

Remote 'Full Establish' operation execution is in progress for
device group 'Rdf1Grp'. Please wait...
Remote 'Full Establish' operation successfully initiated for device
group
'Rdf1Grp'.
```

- The `symmir verify` command checks at 30-second intervals (`-i 30`) to determine when the remote BCV pair on Hop 1 reaches the Synchronized state.

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

None of the devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.

None of the devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.
.......................................................................................

All of the devices in group 'Rdf1Grp' are in the 'Synchronized or Restored' state.
```

- The `symmir query` command confirms that the BCV pair (R2 with the RBCV) on Hop 1 is synchronized.

```
symmir -g Rdf1Grp query -rdf

Device Group (DG) Name: Rdf1Grp
DG's Type            : RDF1
DG's Symmetrix ID    : 000187720758
Remote Symmetrix ID  : 000187900035


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


     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001         047B      0 RBCV001                02BB *        0 Synchronized

Total                -------                               -------
  MB(s)                  0.0                                   0.0


Legend:

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

- While synchronized as a BCV pair, the R1 BCV on Hop 1 is temporarily suspended from copying data to its SRDF-paired R2 device on Hop 2. To resume mirroring between the two, split the Hop-1 BCV pair and then re-establish the Hop-1 BCV with its remote R2 mirror. The following command performs a consistent split on the Hop-1 BCV pair.

```
symmir -g Rdf1Grp split -consistent -noprompt -rdf

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

Remote 'Split' operation successfully executed for device group
'Rdf1Grp'.
```

- This `symrdf establish` command re-establishes the Hop-1 BCV with its remote R2 mirror on the Hop-2 Symmetrix array. The `-rbcv` option specifies that this operation involves the Hop-1 BCV.

```
symrdf -g Rdf1Grp establish -rbcv -noprompt

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

    Suspend RDF link(s).......................................Done.
    Resume RDF link(s)........................................Started.
    Merge device track tables between source and target.......Started.
    Device: 02BB ............................................ Merged.
    Merge device track tables between source and target.......Done.
    Resume RDF link(s)........................................Done.

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

- The `symrdf query` command with the `-rbcv` option shows that R1 BCV on Hop 1 is in the process of synchronizing with its SRDF-paired R2 device on Hop 2.

```
symrdf -g Rdf1Grp -rbcv query

Device Group (DG) Name            : Rdf1Grp
DG's Type                         : RDF1
DG's Symmetrix ID                 : 000187720758
RBCV's Symmetrix ID               : 000187900035
                     R E M O T E      S Y M M E T R I X


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

RBCV001 02BB RW       0    23714 RW 0260 NR       0        0 S..   SyncInProg

Total          -------- --------          -------- --------
  MB(s)             0.0    741.1               0.0      0.0
Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

- The `symrdf verify` command checks at 30-second intervals (`-i 30`) to determine when the Hop-1 BCV and its remote R2 mirror reaches the Synchronized state.

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

NONE of the mirrored pairs are in the 'Synchronized' state.

NONE of the mirrored pairs are in the 'Synchronized' state.
.........................................................................

All devices in the RDF group 'Rdf1Grp' are in the 'Synchronized'
state.
```

- The `symrdf suspend` command suspends mirroring between the Hop-1 BCV and its remote R2 mirror on Hop 2.

```
symrdf -g Rdf1Grp -rbcv suspend -noprompt

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

    Suspend RDF link(s).......................................Done.

The RDF 'Suspend' operation successfully executed for
device group 'Rdf1Grp'.

And then you use the existing material from here!
```

- Now that the R2 device on Hop 2 holds the data, it is possible to clone that data to a BCV on Hop 2. The `symclone establish` command clones a copy to the Hop-2 BCV target (RRBCV001). You specify the Hop-1 BCV as the source, and the data is copied from that BCV's paired R2 device on Hop 2. The `-rrbcv` option specifies that this operation involves the Hop-2 BCV.

```
symclone -g Rdf1Grp establish -full -noprompt RBCV001 sym ld RRBCV001 -rrbcv

'Clone Full Establish' operation execution is in progress for device 'RBCV001'
paired with target device 'RRBCV001' in
device group 'Rdf1Grp'. Please wait...

'Clone Full Establish' operation successfully executed for device 'RBCV001'
in group 'Rdf1Grp' paired with target device 'RRBCV001'.
```

- The `symclone query` command shows the progress of the Hop-2 copy operation.

```
symclone -g Rdf1Grp -rrbcv query

Device Group (DG) Name: Rdf1Grp
DG's Type             : RDF1
DG's Symmetrix ID     : 000187720758
Remote Symmetrix ID   : 000187900041


          Source Device                        Target Device           State      Copy
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                    Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----

RBCV001   0260    115459        0 RRBCV001  022F        0 XXX. CopyInProg    16

Total            --------
  MB(s)            3608.1

Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

- The `symclone verify` command checks at 30-second intervals (`-i 30`) to determine when the Hop-2 clone pair reaches the Copied state.

```
symclone -g Rdf1Grp -rrbcv verify -i 30

None of the devices in group 'Rdf1Grp' are in the 'Copied' state.

......................................................................

All devices in group 'Rdf1Grp' have a paired TGT device that is in
the 'Copied' state.
```

- The `symclone terminate` command terminates the Hop-2 clone session.

```
symclone -g Rdf1Grp terminate -noprompt RBCV001 sym ld RRBCV001 -rrbcv

'Terminate' operation execution is in progress for device 'RBCV001'
paired with target device 'RRBCV001' in
device group 'Rdf1Grp'. Please wait...

'Terminate' operation successfully executed for device 'RBCV001'
in group 'Rdf1Grp' paired with target device 'RRBCV001'.
```

Example 12: Using a clone from a clone target

This example was performed using Solutions Enabler Version 7.0 TimeFinder/Clone, commands. The Symmetrix array is running Enginuity 5874, therefore the TimeFinder operation is a TimeFinder/Clone Emulation session, which is ultimately a clone session.

The device BCV001 is the target device of the TimeFinder/Clone Emulation split session. Since a clone session can be created from an existing clone target, another clone session can be created, which uses the BCV001 as the source device.

symmir query -g TestDg -multi

```
Device Group (DG) Name: TestDg
DG's Type             : REGULAR
DG's Symmetrix ID     : 000192600090


     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001         00A8      0 BCV001                 02B8 *        0 Split

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

symclone create -g TestDg -bcv -tgt

```
'Create' operation execution is in progress for
device group '74'. Please wait...


SELECTING Source devices in the group:

  Device: 02B8 [SELECTED]

SELECTING Target devices in the group:

  Device: 00A9 [SELECTED]

PAIRING of Source and Target devices:

  Devices: 02B8(S) - 00A9(T) [PAIRED]

STARTING a Clone 'CREATE' operation.

'Create' operation successfully executed for device group
'TestDg'.


symclone query -multi -bcv -g TestDg


Device Group (DG) Name: TestDg
DG's Type             : REGULAR
DG's Symmetrix ID     : 000192600090

         Source Device                   Target Device            State     Copy
```

```
--------------------------------- ---------------------------- ------------ ----
               Protected Modified                Modified
Logical    Sym  Tracks    Tracks   Logical    Sym  Tracks    CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------ ----
BCV001     02B8     33000        0 TGT001     00A9        0 .X.. Created       0

Total           -------- --------                -------- 
  Track(s)         33000        0                       0
  MB(s)           2062.5      0.0                     0.0
```

6

Performing TimeFinder/Snap Operations

This chapter provides examples on creating space-saving, instant point-in-time copies of volume data on multiple target devices from a single source device.

The following examples illustrate TimeFinder/Snap functionality in versions of EMC Solutions Enabler up through Version 6.0 running on Symmetrix arrays using Enginuity release levels 5x71, 5x70, or earlier.

Example 1: Snapping to VDEVs and restoring to the original SRC device

This example was performed using Solutions Enabler Version 5.4. The hardware setup consists of a Symmetrix array (sid 206) connected to a Windows 2000 host. The source of the snap is a standard device (76). The targets are three virtual devices (4F, 50, and 51). At different points in time, each virtual device is paired with the standard device and is the target of snap operation. A device group is used to control these three snap pairs.

After the last snap completes, data is restored to the standard device from one of the virtual devices.

- The `symdev list` command displays devices on the Symmetrix array that the example can use as the target virtual devices (VDEV). Devices with the SV attribute have been configured as SAVE devices. To list VDEVs only, use `symdev list -vdev`. The ellipsis (. . .) indicates where superfluous output was omitted or truncated.

```
symdev list

Symmetrix ID: 000000006206

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

0000 Not Visible             ???:? 01A:C0  BCV           Asst'd       RW    8632
0001 Not Visible             ???:? 16C:D0  BCV           Asst'd       RW    8632
0002 Not Visible             ???:? 01B:D0  BCV           Asst'd       RW    8632
0003 Not Visible             ???:? 16D:C0  BCV           Asst'd       RW    8632
................................................................................
0048 Not Visible             ???:? ???:??  VDEV          N/Grp'd      RW    8632
0049 Not Visible             ???:? ???:??  VDEV          N/Grp'd      RW    8632
004A Not Visible             ???:? ???:??  VDEV          N/Grp'd      RW    8632
004B Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
004C Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
004D Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
004E Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
004F Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
0050 Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
0051 Not Visible             ???:? ???:??  VDEV          N/Grp'd      NR    8632
0052 Not Visible             ???:? 16B:D1  2-Way Mir     N/A     (SV) RW    8632
0053 Not Visible             ???:? 01C:C0  2-Way Mir     N/A     (SV) RW    8632
0054 Not Visible             ???:? 16B:C0  2-Way Mir     N/A     (SV) RW    8632
................................................................................
```

- The `symdg` command creates a device group named 3vdevs. The `set` command defines the environment variable `SYMCLI_DG` so that you can omit the device group name in subsequent commands. The first `symld` command adds the source standard device 76 to the device group. The second `symld` command adds the three target virtual devices (4F, 50, and 51) to the device group.

```
symdg create 3vdevs -type regular
set SYMCLI_DG=3vdevs
symld add dev 76
symld addall dev -range 4F:51 -vdev
```

- The `symld list` command displays the devices in the device group 3vdevs and that the virtual devices do not yet have copy sessions with the standard device (indicated by their status of NotCreated).

```
symld list

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

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

DEV001       DRIVE29           0076  04A:0 16C:D2  2-Way Mir         RW   8632

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

Symmetrix Virtual Devices associated with this dg:

            VDEV Device                       Source Device           Status
----------------------------------- ---------------------------- -------------

Logical          Sym     Att.     Sts Logical         Sym          SRC <=> VDEV
----------------------------------- ---------------------------- -------------

VDEV001          004F             NR  N/A             N/A          NotCreated
VDEV002          0050             NR  N/A             N/A          NotCreated
VDEV003          0051             NR  N/A             N/A          NotCreated

Legend for VDEV devices:

(+): VDEV is paired with a member device.
(-): VDEV is paired with a non-member device.
```

- The `symsnap list` command indicates that there are no copy sessions running on the Symmetrix array.

```
symsnap list

Symmetrix ID: 000000006206

No Snap sessions found
```

- The `symsnap list` command with the `-savedevs` option displays any SAVE devices that have been configured on this Symmetrix array to hold data for virtual devices. This display indicates there has not yet been any virtual device activity that caused tracks of data to be copied to these SAVE devices.

`symsnap list -savedevs`

```
Symmetrix ID: 000000006206

                  Snap Save Devices
----------------------------------------------------
      Device     Total       Used       Free  Percent
Sym     Type    Tracks     Tracks     Tracks     Full
----------------------------------------------------
0052     FBA    276210          0     276210       0%
0053     FBA    276210          0     276210       0%
0054     FBA    276210          0     276210       0%
0055     FBA    276210          0     276210       0%
0056     FBA    276210          0     276210       0%
0057     FBA    276210          0     276210       0%
0058     FBA    276210          0     276210       0%
0059     FBA    276210          0     276210       0%
005A     FBA    276210          0     276210       0%
005B     FBA    276210          0     276210       0%

Total         --------- --------- ---------     ----
  Tracks        2762100         0   2762100       0%
  MB(s)         86315.6       0.0   86315.6
```

- The `symsnap create` command creates the snap pair session, sets up a track protection bitmap for each of the paired devices, and places the snap pair in the Created state.

`symsnap create -v DEV001 vdev ld VDEV001 -noprompt`

```
'Snap Create' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV001' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 004F(T) [PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Snap Create' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV001'.
```

- The `symsnap activate` command places the snap pair in the CopyOnWrite (copy-on-first-write) state and the target virtual device in the Read/Write (RW) state. The actual copying of data is deferred until you modify tracks on either the source or the target.

```
symsnap activate -v DEV001 vdev ld VDEV001 -noprompt

'Snap Activate' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV001' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 004F(T) [PAIRED]

STARTING a Snap 'ACTIVATE' operation.

The Snap 'ACTIVATE' operation SUCCEEDED.

'Snap Activate' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV001'.
```

- The `symsnap query` command confirms that the first snap pair is in the copy-on-first-write state.

```
symsnap query

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device           Target Device                  State     Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                      Changed
Logical    Sym   Tracks     Logical    Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     0076      276210 VDEV001    004F    X          0 CopyOnWrite    0

Total                --------                    ----------
  MB(s)                8631.6                          0.0

Legend:

               (G): The Target device is associated with this group.
```

- Although not shown, the example now writes some data to the source device.
- Another `symsnap create` command creates a second copy session for DEV001. This time the source device is paired with the target virtual device VDEV002.

```
symsnap create -v DEV001 vdev ld VDEV002 -noprompt

'Snap Create' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV002' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0050(T) [PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Snap Create' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV002'.
```

- This `symsnap activate` command places the new snap pair in the copy-on-first-write state.

```
symsnap activate -v DEV001 vdev ld VDEV002 -noprompt

'Snap Activate' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV002' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0050(T) [PAIRED]

STARTING a Snap 'ACTIVATE' operation.

The Snap 'ACTIVATE' operation SUCCEEDED.

'Snap Activate' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV002'.
```

- The `symsnap query` command with the `-multi` option displays both copy sessions. Note that the write activity on the source prior to activating this second copy session caused the original data on 2607 tracks to be copied to SAVE devices to which VDEV001 now points. However, the amount of data written was too small to cause the Copy column for VDEV001 to rise above 0%.

```
symsnap query -multi

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device             Target Device                      State      Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                                Changed
Logical    Sym   Tracks     Logical    Sym      G     Tracks    SRC <=> TGT   (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     0076      276210 VDEV002    0050     X            0 CopyOnWrite    0
                     273603 VDEV001    004F     X         2607 CopyOnWrite    0

Total                --------                       ----------
  Track(s)             549813                             2607
  MB(s)               17181.5                             81.5

Legend:

(G): The Target device is associated with this group.
```

- Although not shown, the example now writes again to the same tracks on the source device.

  Another `symsnap create` command creates a third copy session for DEV001.

`symsnap create -v DEV001 vdev ld VDEV003 -noprompt`

```
'Snap Create' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV003' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0051(T) [PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Snap Create' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV003'.
```

- This `symsnap activate` command places the third snap pair in the copy-on-first-write state.

`symsnap activate -v DEV001 vdev ld VDEV003 -noprompt`

```
'Snap Activate' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV003' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0051(T) [PAIRED]

STARTING a Snap 'ACTIVATE' operation.

The Snap 'ACTIVATE' operation SUCCEEDED.

'Snap Activate' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV003'.
```

- The `symsnap query` command with the `-multi` option displays all three copy sessions.

```
symsnap query -multi

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device          Target Device                State     Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                       Changed
Logical    Sym  Tracks    Logical   Sym     G    Tracks   SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     0076     276210 VDEV003   0051    X          0 CopyOnWrite    0
                    273902 VDEV002   0050    X       2308 CopyOnWrite    0
                    273603 VDEV001   004F    X       2607 CopyOnWrite    0

Total            --------                     ----------
  Track(s)          823715                          4915
  MB(s)            25741.0                         153.7

Legend:

(G): The Target device is associated with this group.
```

- Prior to performing a restore operation to the original source device, you should stop all activity on the file system. You can accomplish this task by unmounting the file system.

- The `symsnap restore` command initiates an incremental restore operation that restores to DEV001 the original track data that was copied to a SAVE device during the DEV001/VDEV002 copy session.

```
symsnap restore DEV001 vdev ld VDEV003

'Snap Incremental Restore' operation execution is in progress for device 'DEV001'
paired with target device 'VDEV003' in device group '3vdevs'. Please wait...

'Snap Incremental Restore' operation successfully initiated for device 'DEV001'
in group '3vdevs' paired with target device 'VDEV003'.
```

- The `symsnap query` command with the `-restore` option shows the restore in progress.

```
symsnap query -restore

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device          Target Device     Virtual Device    State      Copy
------------------------ --------------------- -------------- ------------ ----
               Protected                  Indirect
Logical    Sym  Tracks    Logical Sym  Tracks   Logical    Sym  SRC <=> TGT  (%)
------------------------ --------------------- -------------- ------------ ----

DEV001     0076         0 VDEV003 0051      155 VDEV003    0051 RestInProg    99

Total           --------                -------
  MB(s)              0.0                    4.8
```

- The `symsnap query` command with the `-multi` option displays the three original copy sessions plus the restore session.

```
symsnap query -multi

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device          Target Device                  State      Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                       Changed
Logical    Sym   Tracks    Logical    Sym     G   Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----
DEV001     0076        0 VDEV003    0051     X         0 Restored     100
                  276210 VDEV003    0051     X         0 CopyOnWrite    0
                  273902 VDEV002    0050     X      2308 CopyOnWrite    0
                  273603 VDEV001    004F     X      2607 CopyOnWrite    0

Total            --------                     ----------
  Track(s)         823715                           4915
  MB(s)           25741.0                          153.7

Legend:

(G): The Target device is associated with this group.
```

- Once a restore operation is in progress (or complete), you can import the volume group and mount the file system.

  Once the restore operation is complete, you should terminate the restore session. The following command terminates the DEV001/VDEV003 restore session, but does not affect its copy session.

```
symsnap terminate -v DEV001 vdev ld VDEV003 -restored

Execute 'Snap Terminate' operation for device 'DEV001' in device group '3vdevs'
(y/[n]) ? y

'Snap Terminate' operation execution is in progress for device 'DEV001' paired
with target device 'VDEV003' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0051(T) [PAIRED]

STARTING a Snap 'TERMINATE' operation.

The Snap 'TERMINATE' operation SUCCEEDED.

'Snap Terminate' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'VDEV003'.
```

- When a restore operation is initiated, both the virtual device and the original source device (target of the restore) are set to Not Ready. Once the restore operation is underway, the source device is automatically set to Ready again. However, if you need to access the virtual device after the restore completes, you must use the `symld ready` command to set it to Ready again. The following command sets VDEV003 to Ready.

  ```
  symld ready VDEV003 -noprompt -vdev

  'Ready' device operation successfully completed for the device.
  ```

Example 2: Snapping to VDEVs and restoring to a BCV

This example continues Example 1 but adds to the device group a BCV device (24) that is used as the target of the restore operation. The BCV is established with the source standard device (76) and thus has a pairing relationship with the same standard device that is the source for the three target virtual devices (4F, 50, and 51). After the snap completes and data is written to the source, original change-track data is restored to the BCV.

- The `symsnap query` command with the `-multi` option now displays the three copy sessions. The restore session from Example 1 has been terminated.

```
symsnap query -multi

Device Group (DG) Name: 3vdevs
DG's Type            : REGULAR
DG's Symmetrix ID    : 000000006206

      Source Device          Target Device                State      Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                        Changed
Logical    Sym  Tracks    Logical    Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     0076     276210 VDEV003   0051    X          0 CopyOnWrite    0
                    273902 VDEV002   0050    X       2308 CopyOnWrite    0
                    273603 VDEV001   004F    X       2607 CopyOnWrite    0

Total                --------                 ----------
  Track(s)            823715                        4915
  MB(s)              25741.0                       153.7

Legend:

(G): The Target device is associated with this group.
```

- The `symbcv associate` command associates the BCV device (24) with the 3vdevs device group.

  `symbcv associate dev 24`

- The `symmir establish` command initiates a full establish operation on the BCV pair (76/24) in the device group 3vdevs.

```
symmir establish -full -noprompt

'Full Establish' operation execution is in progress for device 'DEV001' paired
with BCV device 'BCV001' in device group '3vdevs'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in group
'3vdevs' paired with BCV device 'BCV001'.
```

- The `symmir verify` command checks the establish operation every five seconds and verifies when the operation is complete. That is, DEV001 has been fully copied to BCV001.

```
symmir verify DEV001 -i 5

Device 'DEV001' in group '3vdevs' is NOT in the 'Synchronized or Restored' state.

Device 'DEV001' in group '3vdevs' is in the 'Synchronized or Restored' state.
```

- The `symld list` command shows the addition of the BCV to the 3vdevs device group and that the BCV pair is in the Synchronized state.

```
symld list

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

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

DEV001       DRIVE29            0076 + 04A:0 16C:D2  2-Way Mir           RW    8632

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                    Standard Device          Status
------------------------------------ --------------------------- ------------
                                Inv.                        Inv.
Logical          Sym   RDF Att. Tracks Logical         Sym   Tracks BCV <=> STD
------------------------------------ --------------------------- ------------

BCV001           0024         +       0 DEV001          0076       0 Synchronized

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

Legend for BCV devices:

(+): BCV is paired with a member STD device.
(-): BCV is paired with a non-member STD device.
Symmetrix Virtual Devices associated with this dg:

            VDEV Device                      Source Device           Status
----------------------------------- --------------------------- -------------

Logical          Sym      Att.    Sts Logical          Sym            SRC <=> VDEV
----------------------------------- --------------------------- -------------

VDEV001          004F         +    RW   DEV001           0076           CopyOnWrite
VDEV002          0050         +    RW   DEV001           0076           CopyOnWrite
VDEV003          0051         +    RW   DEV001           0076           CopyOnWrite

Legend for VDEV devices:

(+): VDEV is paired with a member device.
(-): VDEV is paired with a non-member device.
```

- The `symmir split` command performs an instant split on the BCV pair.

```
symmir split -instant -noprompt

'Split' operation execution is in progress for device 'DEV001' paired with BCV
device 'BCV001' in device group '3vdevs'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group '3vdevs'
paired with BCV device 'BCV001'.
```

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

```
symmir verify -split -bg -i 5

None of the devices in group '3vdevs' have finished splitting in the background.

None of the devices in group '3vdevs' have finished splitting in the background.

None of the devices in group '3vdevs' have finished splitting in the background.

All devices in group '3vdevs' have finished splitting in the background.
```

- The `symld list` command confirms that the BCV pair is now in the Split state and ready to be restored from the virtual device (VDEV001).

```
symld list

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

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

DEV001       DRIVE29           0076 + 04A:0 16C:D2  2-Way Mir          RW    8632

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                      Standard Device          Status
------------------------------------ --------------------------- ------------
                              Inv.                            Inv.
Logical          Sym   RDF Att. Tracks Logical            Sym    Tracks BCV <=> STD
------------------------------------ --------------------------- ------------

BCV001           0024          +       0 DEV001           0076       0 Split

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

Legend for BCV devices:

(+): BCV is paired with a member STD device.
(-): BCV is paired with a non-member STD device.

Symmetrix Virtual Devices associated with this dg:
```

```
              VDEV Device                              Source Device            Status
----------------------------------- ---------------------------- -------------

Logical           Sym      Att.     Sts Logical           Sym           SRC <=> VDEV
----------------------------------- ---------------------------- -------------

VDEV001           004F         +    RW  DEV001            0076          CopyOnWrite
VDEV002           0050         +    RW  DEV001            0076          CopyOnWrite
VDEV003           0051         +    RW  DEV001            0076          CopyOnWrite

Legend for VDEV devices:

(+): VDEV is paired with a member device.
(-): VDEV is paired with a non-member device.
```

- Prior to performing a restore operation, you should stop all activity on the file system. You can accomplish this task by unmounting the file system.

  The `symsnap restore` command initiates an incremental restore operation that restores to BCV001 the original track data that was copied to SAVE devices during the DEV001/VDEV001 copy session.

```
symsnap restore BCV001 vdev ld VDEV001 -noprompt

'Snap Incremental Restore' operation execution is in progress for device 'BCV001'
paired with target device 'VDEV001' in device group '3vdevs'. Please wait...

'Snap Incremental Restore' operation successfully initiated for device 'BCV001'
in group '3vdevs' paired with target device 'VDEV001'.
```

- The `symsnap query` command with the `-restore` option shows that the restore operation is complete (state is Restored).

```
symsnap query -restore

Device Group (DG) Name: 3vdevs
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device          Target Device      Virtual Device     State      Copy
------------------------ --------------------- -------------- ------------ ----
               Protected               Indirect
Logical    Sym  Tracks     Logical  Sym  Tracks   Logical    Sym  SRC <=> TGT  (%)
------------------------ --------------------- -------------- ------------ ----

DEV001     0076          0 BCV001   0024        0 VDEV001    004F Restored     100

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

- Once a restore operation is in progress (or complete), you can import the volume group and mount the file system.

  The following command terminates the DEV001/BCV001 restore session.

```
symsnap terminate -v DEV001 bcv ld BCV001

Execute 'Snap Terminate' operation for device 'DEV001' in device group '3vdevs'
(y/[n]) ? y

'Snap Terminate' operation execution is in progress for device 'DEV001' paired
with target device 'BCV001' in device group '3vdevs'. Please wait...

PAIRING of Source and Target devices:

  Devices: 0076(S) - 0024(T) [PAIRED]

STARTING a Snap 'TERMINATE' operation.

The Snap 'TERMINATE' operation SUCCEEDED.

'Snap Terminate' operation successfully executed for device 'DEV001' in group
'3vdevs' paired with target device 'BCV001'.
```

- When a restore operation is initiated, both the virtual device and the target device are set to Not Ready. Once the restore operation is underway, the target device is automatically set to Ready again. However, if you need to access the virtual device after the restore completes, you must use the `symld ready` command to set it to Ready again. The following command sets VDEV001 to Ready.

  ```
  symld ready VDEV001 -noprompt -vdev

  'Ready' device operation successfully completed for the device.
  ```

Example 3: Snapping to VDEVs and restoring to separate STD devices

This example was performed using Solutions Enabler Version 5.4. The hardware setup consists of a Symmetrix array (sid 6206) connected to a Windows 2000 host. The source of the snap is three standard devices (6D, 6E, and 6F). Two virtual devices are paired with each of the three source devices. The six virtual devices are VDEV devices 48 through 4D.

After the snap completes, the example restores data from the virtual devices to separate standard devices (70, 71, and 72).

- The `symdev list` command displays devices on the Symmetrix array that the example can use as the virtual devices (VDEV). The six chosen are VDEV devices 48 through 4D. Devices with the SV attribute have been configured as SAVE devices. To list VDEVs only, use `symdev list -vdev`. The ellipsis (......) indicates where superfluous output was omitted or truncated.

```
symdev list

Symmetrix ID: 000000006206

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

0000 Not Visible             ???:? 01A:C0  BCV            Asst'd       RW    8632
0001 Not Visible             ???:? 16C:D0  BCV            Asst'd       RW    8632
0002 Not Visible             ???:? 01B:D0  BCV            Asst'd       RW    8632
..............................................................................
0011 Not Visible             ???:? 16C:D0  BCV            N/Asst'd     RW    8632
0012 Not Visible             ???:? 01B:D0  BCV            N/Asst'd     RW    8632
0013 Not Visible             ???:? 16D:C0  BCV            N/Asst'd     RW    8632
0014 Not Visible             ???:? 01A:D1  BCV            N/Asst'd     NR    8632
..............................................................................
0048 Not Visible             ???:? ???:??  VDEV           N/Grp'd      RW    8632
0049 Not Visible             ???:? ???:??  VDEV           N/Grp'd      RW    8632
004A Not Visible             ???:? ???:??  VDEV           N/Grp'd      RW    8632
004B Not Visible             ???:? ???:??  VDEV           N/Grp'd      NR    8632
004C Not Visible             ???:? ???:??  VDEV           N/Grp'd      NR    8632
004D Not Visible             ???:? ???:??  VDEV           N/Grp'd      NR    8632
004E Not Visible             ???:? ???:??  VDEV           N/Grp'd      NR    8632
004F Not Visible             ???:? ???:??  VDEV           Grp'd        NR    8632
0050 Not Visible             ???:? ???:??  VDEV           Grp'd        NR    8632
0051 Not Visible             ???:? ???:??  VDEV           Grp'd        NR    8632
0052 Not Visible             ???:? 16B:D1  2-Way Mir      N/A     (SV) RW    8632
0053 Not Visible             ???:? 01C:C0  2-Way Mir      N/A     (SV) RW    8632
..............................................................................
```

- The `symdg create` command creates a device group named vgroup.

  `symdg create vgroup -type regular`

- The following `symld` command adds to the device group standard devices 6D, 6E, and 6F to be sources for the virtual device copies.

  `symld -g vgroup addall dev -range 6D:6F`

- The following `symld` command adds virtual devices 48, 49, 4A, 4B, 4C, and 4D to the device group. Two virtual devices will be paired with each of the three source devices.

  symld -g vgroup addall dev -range 48:4D -vdev

- Although not shown, the example creates a table on the new database and inserts 10 rows of data.

  The `symsnap create` command creates snap pairs from the three source standard devices and the first three virtual devices.

```
symsnap -g vgroup -sid 6206 create -v -nop DEV001 vdev ld VDEV001
                                           DEV002 vdev ld VDEV002
                                           DEV003 vdev ld VDEV003

'Snap Create' operation execution is in progress for the device list in device
group 'vgroup'. Please wait...

SELECTING the list of Source devices in the group:

  Device: 006D [SELECTED]
  Device: 006E [SELECTED]
  Device: 006F [SELECTED]

SELECTING the list of Target devices in the group:

  Device: 0048 [SELECTED]
  Device: 0049 [SELECTED]
  Device: 004A [SELECTED]

PAIRING of Source and Target devices:

  Devices: 006D(S) - 0048(T) [PAIRED]
  Devices: 006E(S) - 0049(T) [PAIRED]
  Devices: 006F(S) - 004A(T) [PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Snap Create' operation successfully executed for the device list in device
group 'vgroup'.
```

- The `symsnap query` command displays the snap pairs and their Created state.

```
symsnap -g vgroup query

Device Group (DG) Name: vgroup
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000006206

      Source Device            Target Device                  State     Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                       Changed
Logical    Sym  Tracks    Logical   Sym     G   Tracks   SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     006D    276210 VDEV001   0048    X          0 Created        0
DEV002     006E    276210 VDEV002   0049    X          0 Created        0
DEV003     006F    276210 VDEV003   004A    X          0 Created        0

Total              --------                   ----------
  Track(s)           828630                            0
  MB(s)             25894.7                          0.0
Legend:
(G): The Target device is associated with this group.
```

- The `symsnap activate` command places each snap pair in the copy-on-first-write state.

```
symsnap -g vgroup -sid 6206 activate -v -nop DEV001 vdev ld VDEV001
                                             DEV002 vdev ld VDEV002
                                             DEV003 vdev ld VDEV003

'Snap Activate' operation execution is in progress for the device list in device
group 'vgroup'. Please wait...

SELECTING the list of Source devices in the group:

  Device: 006D [SELECTED]
  Device: 006E [SELECTED]
  Device: 006F [SELECTED]

SELECTING the list of Target devices in the group:

  Device: 0048 [SELECTED]
  Device: 0049 [SELECTED]
  Device: 004A [SELECTED]

PAIRING of Source and Target devices:

  Devices: 006D(S) - 0048(T) [PAIRED]
  Devices: 006E(S) - 0049(T) [PAIRED]
  Devices: 006F(S) - 004A(T) [PAIRED]

STARTING a Snap 'ACTIVATE' operation.

The Snap 'ACTIVATE' operation SUCCEEDED.

'Snap Activate' operation successfully executed for the device list in device
group 'vgroup'.
```

- Another query for the device group shows that the snap pairs are now in the copy-on-first-write state.

```
symsnap -g vgroup query

Device Group (DG) Name: vgroup
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device           Target Device                  State       Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                        Changed
Logical    Sym   Tracks    Logical    Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     006D    276210 VDEV001    0048     X          0 CopyOnWrite    0
DEV002     006E    276210 VDEV002    0049     X          0 CopyOnWrite    0
DEV003     006F    276210 VDEV003    004A     X          0 CopyOnWrite    0

Total            --------                     ----------
  Track(s)         828630                              0
  MB(s)           25894.7                            0.0

Legend:

(G): The Target device is associated with this group.
```

- Although not shown, the example now writes some data to the source, inserting 100k rows of data into the database table.

  The `symsnap create` command creates a second set of snap pairs, matching the last three virtual devices with the three source standard devices.

```
symsnap -g vgroup -sid 6206 create -v -nop DEV001 vdev ld VDEV004
                                           DEV002 vdev ld VDEV005
                                           DEV003 vdev ld VDEV006

'Snap Create' operation execution is in progress for the device list in device
group 'vgroup'. Please wait...

SELECTING the list of Source devices in the group:

  Device: 006D [SELECTED]
  Device: 006E [SELECTED]
  Device: 006F [SELECTED]

SELECTING the list of Target devices in the group:

  Device: 004B [SELECTED]
  Device: 004C [SELECTED]
  Device: 004D [SELECTED]

PAIRING of Source and Target devices:

  Devices: 006D(S) - 004B(T) [PAIRED]
  Devices: 006E(S) - 004C(T) [PAIRED]
  Devices: 006F(S) - 004D(T) [PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Snap Create' operation successfully executed for the device list in device
group 'vgroup'.
```

- The `symsnap activate` command activates the new snap pair copy sessions.

```
symsnap -g vgroup -sid 6206 activate -v -nop DEV001 vdev ld VDEV004
                                             DEV002 vdev ld VDEV005
                                             DEV003 vdev ld VDEV006

'Snap Activate' operation execution is in progress for the device list in device
group 'vgroup'. Please wait...

SELECTING the list of Source devices in the group:

  Device: 006D [SELECTED]
  Device: 006E [SELECTED]
  Device: 006F [SELECTED]

SELECTING the list of Target devices in the group:

  Device: 004B [SELECTED]
  Device: 004C [SELECTED]
  Device: 004D [SELECTED]

PAIRING of Source and Target devices:

  Devices: 006D(S) - 004B(T) [PAIRED]
  Devices: 006E(S) - 004C(T) [PAIRED]
  Devices: 006F(S) - 004D(T) [PAIRED]

STARTING a Snap 'ACTIVATE' operation.
```

```
The Snap 'ACTIVATE' operation SUCCEEDED.

'Snap Activate' operation successfully executed for the device list in device
group 'vgroup'.
```

- The `symsnap query -multi` command shows multiple copy sessions for a source device paired with multiple virtual devices. Note that each source device has current copy sessions with two virtual devices. For example, the source device 6D has copy sessions with virtual devices 48 and 4B. Also, virtual devices from the first group now point to changed tracks that resulted from previous writes to the database.

symsnap -g vgroup query -multi

```
Device Group (DG) Name: vgroup
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000006206

      Source Device          Target Device                  State      Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                                Changed
Logical    Sym   Tracks     Logical    Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     006D     276210 VDEV004    004B     X           0 CopyOnWrite    0
                    275457 VDEV001    0048     X         753 CopyOnWrite    0
DEV002     006E     276210 VDEV005    004C     X           0 CopyOnWrite    0
                    275460 VDEV002    0049     X         750 CopyOnWrite    0
DEV003     006F     276210 VDEV006    004D     X           0 CopyOnWrite    0
                    275461 VDEV003    004A     X         749 CopyOnWrite    0

Total              --------                     ----------
  Track(s)          1655007                           2253
  MB(s)             51719.0                           70.4

Legend:

(G): The Target device is associated with this group.
```

- The `symld` command adds to the device group those standard devices (70, 71, and 72) that will be targets of the restore operation.

  symld -g vgroup addall dev -range 70:72

- The `symsnap restore` command with the `-full` option initiates the restore operation to devices 70, 71, and 72 from the first three virtual devices. The targets of the restore receive change-track data pointed to by the virtual devices as well as unchanged track data from the original source devices. That combination results in a full restore.

```
symsnap -g vgroup -sid 6206 restore -full -nop DEV004 vdev ld VDEV001
                                               DEV005 vdev ld VDEV002
                                               DEV006 vdev ld VDEV003

'Snap Full Restore' operation execution is in progress for the device list in
device group 'vgroup'. Please wait...

'Snap Full Restore' operation successfully initiated for the device list in
device group 'vgroup'.
```

- The `symsnap query` command with the `-restore` option shows that the restore operation is in progress. The target devices receive change-track data pointed to by the virtual devices as well as unchanged track data from the original source devices. That combination results in a full restore.

symsnap -g vgroup query -restore

```
Device Group (DG) Name: vgroup
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000006206

      Source Device          Target Device      Virtual Device    State      Copy
------------------------ --------------------- -------------- ------------ ----
                Protected               Indirect
Logical     Sym  Tracks     Logical  Sym  Tracks    Logical     Sym  SRC <=> TGT  (%)
------------------------ --------------------- -------------- ------------ ----

DEV001     006D    253169 DEV004   0070    253169 VDEV001    0048 RestInProg    8
DEV002     006E    252854 DEV005   0071    252854 VDEV002    0049 RestInProg    8
DEV003     006F    243429 DEV006   0072    243429 VDEV003    004A RestInProg   11

Total               --------               -------
  Track(s)            749452                749452
  MB(s)              23420.4               23420.4
```

- The `symsnap query -multi` command now adds the restore sessions to the display.

symsnap -g vgroup query -multi

```
Device Group (DG) Name: vgroup
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000006206

      Source Device          Target Device                   State      Copy
------------------------- ------------------- ---------- ------------ ----
                  Protected                          Changed
Logical     Sym    Tracks    Logical    Sym     G    Tracks   SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     006D          0 VDEV001    0048     X          0 Restored     100
                    276210 VDEV001    0048     X          0 CopyOnWrite    0
                    275457 VDEV004    004B     X        753 CopyOnWrite    0
DEV002     006E          0 VDEV002    0049     X          0 Restored     100
                    276210 VDEV002    0049     X          0 CopyOnWrite    0
                    275460 VDEV005    004C     X        750 CopyOnWrite    0
DEV003     006F          0 VDEV003    004A     X          0 Restored     100
                    276210 VDEV003    004A     X          0 CopyOnWrite    0
                    275461 VDEV006    004D     X        749 CopyOnWrite    0

Total                --------                     ----------
  Track(s)            1655007                           2253
  MB(s)               51719.0                           70.4

Legend:

(G): The Target device is associated with this group.
```

- Before restoring the second set of virtual devices to the same target standard devices, you need to terminate the current restore sessions. The `symsnap terminate` command with the `-restored` option terminates the restore sessions while leaving the copy sessions intact.

```
symsnap -g vgroup -sid 6206 terminate -restored -v DEV001 sym ld DEV004
                                                   DEV002 sym ld DEV005
                                                   DEV003 sym ld DEV006

Execute 'Snap Terminate' operation for the 3 specified device(s) in device group
'vgroup' (y/[n]) ? y

'Snap Terminate' operation execution is in progress for the device list in device
group 'vgroup'. Please wait...

SELECTING the list of Source devices in the group:

  Device: 006D [SELECTED]
  Device: 006E [SELECTED]
  Device: 006F [SELECTED]

SELECTING the list of Target devices in the group:

  Device: 0070 [SELECTED]
  Device: 0071 [SELECTED]
  Device: 0072 [SELECTED]

PAIRING of Source and Target devices:

  Devices: 006D(S) - 0070(T) [PAIRED]
  Devices: 006E(S) - 0071(T) [PAIRED]
  Devices: 006F(S) - 0072(T) [PAIRED]

STARTING a Snap 'TERMINATE' operation.

The Snap 'TERMINATE' operation SUCCEEDED.

'Snap Terminate' operation successfully executed for the device list in device
group 'vgroup'.
```

- The `symsnap query` command shows the remaining copy sessions.

```
symsnap -g vgroup query -multi

Device Group (DG) Name: vgroup
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device           Target Device                State     Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                           Changed
Logical    Sym    Tracks     Logical    Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001     006D     276210 VDEV004    004B     X          0 CopyOnWrite    0
                    275457 VDEV001    0048     X        753 CopyOnWrite    0
DEV002     006E     276210 VDEV005    004C     X          0 CopyOnWrite    0
                    275460 VDEV002    0049     X        750 CopyOnWrite    0
DEV003     006F     276210 VDEV006    004D     X          0 CopyOnWrite    0
                    275461 VDEV003    004A     X        749 CopyOnWrite    0

Total            --------                     ----------
  Track(s)        1655007                           2253
  MB(s)           51719.0                           70.4

Legend:

(G): The Target device is associated with this group.
```

- The `symsnap restore` command with the `-full` option initiates the restore operation to devices 70, 71, and 72 from the *last* three virtual devices.

```
symsnap -g vgroup -sid 6206 restore -full -nop DEV004 vdev ld VDEV004
                                               DEV005 vdev ld VDEV005
                                               DEV006 vdev ld VDEV006

'Snap Full Restore' operation execution is in progress for the device list in
device group 'vgroup'. Please wait...

'Snap Full Restore' operation successfully initiated for the device list in
device group 'vgroup'.
```

- The `symsnap query` command with the `-restore` option shows that the restore operation is in progress. The target devices receive change-track data pointed to by the virtual devices as well as unchanged track data from the original source devices. That combination results in a full restore.

```
symsnap -g vgroup query -restore

Device Group (DG) Name: vgroup
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000006206

      Source Device          Target Device      Virtual Device    State      Copy
------------------------ --------------------- -------------- ------------ ----
               Protected              Indirect
Logical    Sym  Tracks    Logical Sym  Tracks   Logical    Sym  SRC <=> TGT  (%)
------------------------ --------------------- -------------- ------------ ----

DEV001     006D    253169 DEV004   0070   253169 VDEV004    004B RestInProg     8
DEV002     006E    252854 DEV005   0071   252854 VDEV005    004C RestInProg     8
DEV003     006F    243429 DEV006   0072   243429 VDEV006    004D RestInProg    11

Total           --------               -------
  Track(s)        749452                749452
  MB(s)          23420.4               23420.4
```

- Once the restore operation completes, you should terminate the restore sessions as shown earlier. If you need to use the virtual devices again, you need to make them Ready again as described in the previous examples. The following command makes all virtual devices in the device group Ready (RW).

  ```
  symld -g vgroup ready -vdev

  'Ready' device operation successfully completed for the device
  group.
  ```

Example 4: Using a composite group to control snap pairs

This example was performed using Solutions Enabler Version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (sid 35 and 79). A composite group defined on the host contains a set of snap pairs that spans the two Symmetrix arrays. The devices include standard devices and virtual devices. Snap pairs in the composite group are created, activated, and controlled together as a group.

You can create, activate, and control specific snap pairs within the composite group as well as performing control operations on the entire group as shown here. For example, you can restore or terminate one snap session in the group without restoring or terminating the group's other sessions. You can also create a composite group of snap pairs and specify a named save pool, provided that the save pool exists on all Symmetrix arrays in the consistency group.)

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

```
symcg create SNAP -type regular
```

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

```
symcg -cg SNAP addall dev -range 56:6E -sid 35
symcg -cg SNAP addall dev -range 623:3B -sid 79
```

- The following `symcg` commands with the `-vdev` option add to the composite group a range of virtual devices from each of the two source Symmetrix arrays.

```
symcg -cg SNAP addall dev -range EC:104 -sid 35 -vdev
symcg -cg SNAP addall dev -range 598:5B0 -sid 79 -vdev
```

- The `symsnap create` command creates snap pair sessions, pairing standards and virtual devices from the composite group and placing each snap pair in the Created state.

```
symsnap -cg SNAP create -noprompt

'Snap Create' operation execution is in progress for composite group 'SNAP'.
Please wait...

'Snap Create' operation successfully executed for composite group 'SNAP'.
```

- The `symsnap query` command displays the snap pairs and that they are in the Created state. Note that devices 56 and 59 are meta devices (as are their target devices), and only the meta head device is displayed.

`symsnap -cg SNAP query`

```
Composite Group Name       :  SNAP
Composite Group Type       :  REGULAR
Number of Symmetrix Units  :  2
Number of RDF (RA) Groups  :  0

Symmetrix ID                : 000187900035    (Microcode Version: 5670)

       Source Device             Target Device            State     Copy
------------------------- ------------------------------ ------------ ----
                   Prot.                          Changed
Sym                Tracks  Sym               G     Tracks  SRC <=> TGT  (%)
------------------------- ------------------------------ ------------ ----
0056              414180  00EC              X         90  Created        0
0059              276180  00EF              X          0  Created        0
005B              138090  00F1              X          0  Created        0
005C              138090  00F2              X          0  Created        0
005D              138090  00F3              X          0  Created        0
005E              138090  00F4              X          0  Created        0
005F              138090  00F5              X          0  Created        0
0060              138090  00F6              X          0  Created        0
0061              138090  00F7              X          0  Created        0
0062              138090  00F8              X          0  Created        0
0063              138090  00F9              X          0  Created        0
0064              138090  00FA              X          0  Created        0
0065              138090  00FB              X          0  Created        0
0066              138090  00FC              X          0  Created        0
0067              138090  00FD              X          0  Created        0
0068              138090  00FE              X          0  Created        0
0069              138090  00FF              X          0  Created        0
006A              138090  0100              X          0  Created        0
006B              138090  0101              X          0  Created        0
006C              138090  0102              X          0  Created        0
006D              138090  0103              X          0  Created        0
006E              138090  0104              X          0  Created        0

Symmetrix ID                : 000187700079    (Microcode Version: 5670)

       Source Device             Target Device            State     Copy
------------------------- ------------------------------ ------------ ----
                   Prot.                          Changed
Sym                Tracks  Sym               G     Tracks  SRC <=> TGT  (%)
------------------------- ------------------------------ ------------ ----
0023               92100  0598              X          0  Created        0
0024               92100  0599              X          0  Created        0
0025               92100  059A              X          0  Created        0
0026               92100  059B              X          0  Created        0
0027               92100  059C              X          0  Created        0
0028               92100  059D              X          0  Created        0
0029               92100  059E              X          0  Created        0
002A               92100  059F              X          0  Created        0
002B               92100  05A0              X          0  Created        0
002C               92100  05A1              X          0  Created        0
002D               92100  05A2              X          0  Created        0
002E               92100  05A3              X          0  Created        0
002F               92100  05A4              X          0  Created        0
0030               92100  05A5              X          0  Created        0
0031               92100  05A6              X          0  Created        0
0032               92100  05A7              X          0  Created        0
0033               92100  05A8              X          0  Created        0
0034               92100  05A9              X          0  Created        0
```

```
0035                92100 05AA             X           0 Created        0
0036                92100 05AB             X           0 Created        0
0037                92100 05AC             X           0 Created        0
0038                92100 05AD             X           0 Created        0
0039                92100 05AE             X           0 Created        0
003A                92100 05AF             X           0 Created        0
003B                92100 05B0             X           0 Created        0

Total             --------                          ----------
  Track(s)         5754660                                  90
  MB(s)             179833                                   3

Legend:

(G): The Target device is associated with this group.
```

- The `symsnap activate` command places each snap pair in the copy-on-first-write state and the target virtual device in the Read/Write (RW) state. As described previously, the actual copying of data is deferred until you modify tracks on either a source device or a target device. Keep in mind that any subsequent `symsnap` control operations must be performed on the entire composite group.

```
symsnap -cg SNAP activate -noprompt

'Snap Activate' operation execution is in progress for composite group 'SNAP'.
Please wait...

'Snap Activate' operation successfully executed for composite group 'SNAP'.
```

- The `symsnap terminate` command ends all snap sessions in the composite group.

```
symsnap -cg SNAP terminate -noprompt

'Snap Terminate' operation execution is in progress for composite group 'SNAP'.
Please wait...

'Snap Terminate' operation successfully executed for composite group 'SNAP'.
```

7

Attaching, Querying, and Verifying with TimeFinder commands

This chapter provides examples on using the attach, query, and verify operations with TimeFinder family products.

The following examples illustrate legacy TimeFinder/Mirror functionality in versions of EMC Solutions Enabler through Version 5.4 running on Symmetrix arrays using Enginuity 5670 and earlier.

Example 1: Querying a device group

This example shows BCVs with physical device names, signifying that the BCVs are addressable by the host on which the standards are visible. While using a single host is convenient for demonstration purposes, usually a second host is used for accessing BCVs for business continuance tasks such as backups and testing.

- Recall that the `symdev list` and `symbcv list` commands examine records inside the Symmetrix and not information about any specific device group from records stored in the SYMAPI database file. Before creating a device group and adding devices to it, examine standard/BCV information inside the Symmetrix to determine which devices are ungrouped and free from prior relationships. The abbreviated output from the `symdev list` command displays standards and BCVs that do not currently belong to a device group (N/Grp'd or N/Asst'd). The ellipsis (…) indicates where output was omitted.

`symdev list`

```
Symmetrix ID: 000000003264

        Device Name           Directors                  Device
--------------------------- ------------- -------------------------------------
                                                                           Cap
Sym  Physical               SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -------------------------------------
0000 Not Visible            16A:0 01A:C0  Unprotected   N/Grp'd  (M) RW    1031
0001 Not Visible            16A:0 02B:D3  Unprotected   N/Grp'd  (M) RW    1031
0002 Not Visible            ???:? 02A:C0  Unprotected   N/Grp'd      RW     516
0003 Not Visible            16A:0 01B:D3  Unprotected   N/Grp'd  (m) RW       -
0004 Not Visible            16A:0 01B:C0  Unprotected   N/Grp'd  (m) RW       -
0005 Not Visible            ???:? 02A:D3  Unprotected   N/Grp'd      RW     516
...............................................................................
0086 /dev/rdsk/c4t0d2s2     16B:0 01B:C0  2-Way Mir     N/Grp'd      RW     516
0087 /dev/rdsk/c4t0d3s2     16B:0 02B:C0  2-Way Mir     N/Grp'd      RW     516
0088 /dev/rdsk/c4t1d0s2     16B:0 01A:D0  2-Way BCV Mir N/Asst'd     RW     516
0089 /dev/rdsk/c4t1d1s2     16B:0 02A:D0  2-Way BCV Mir N/Asst'd     RW     516
008A /dev/rdsk/c4t1d2s2     16B:0 01B:D0  2-Way BCV Mir N/Asst'd     RW     516
008B /dev/rdsk/c4t1d3s2     16B:0 02B:D0  2-Way BCV Mir N/Asst'd     RW     516
```

- The `symbcv list` command, which lists the BCVs on the left and standards on the right, displays that the devices of interest do have prior pairing relationships (indicated by their Split state).

`symbcv list`

```
Symmetrix ID: 000000003264

             BCV Device                       Standard Device         Status
------------------------------------ --------------------------- ------------
                               Inv.                          Inv.
Physical        Sym  RDF Att. Tracks Physical            Sym  Tracks BCV <=> STD
------------------------------------ --------------------------- ------------
Not Visible     0030      (M)      0 N/A                 N/A       0 NeverEstab
Not Visible     0031      (m)      - N/A                 N/A       - NeverEstab
.............................................................................
c4t1d0s2        0088               0 c4t0d0s2            0084      0 Split
c4t1d1s2        0089               0 c4t0d1s2            0085      0 Split
c4t1d2s2        008A               0 c4t0d2s2            0086      0 Split
c4t1d3s2        008B               0 c4t0d3s2            0087      0 Split
```

- The records inside the Symmetrix show that standard device 84 and BCV device 88 have a pairing relationship. To illustrate the complications that can occur from ignoring that relationship, the following commands build a device group using standard device 84 and BCV 89.

```
symdg create ProdBgrp
symld -g ProdBgrp -sid 3264 add dev 84
symbcv -g ProdBgrp -sid 3264 add dev 89
```

- The `symld list` command examines information about the new device group recorded inside the SYMAPI database, symapi_db.bin, which resides on the host. The host's SYMAPI database is typically located in /var/symapi/db or \Program Files\EMC\SYMAPI\db. Note the minus (-) sign next to the standard as well as the BCV, signifying that each of those devices is paired with a partner that is not part of this device group. In addition, the display indicates that the current partner of BCV 89 is device 85.

```
symld -g ProdBgrp list

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

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

DEV001      /dev/dsk/c4t0d0s2 0084 - 16B:0 01A:C0 2-Way Mir          RW     516

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                    Standard Device          Status
------------------------------------ --------------------------- ------------
                                  Inv.                         Inv.
Logical          Sym  RDF Att. Tracks Logical          Sym   Tracks BCV <=> STD
------------------------------------ --------------------------- ------------
BCV001           0089         -      0                 0085       0 Split

Total                        --------                        --------
  MB(s)                           0.0                             0.0

Legend for BCV devices:

(+): BCV is paired with a member STD device.
(-): BCV is paired with a non-member STD device.
```

- The following `symmir query` of the device group displays the pairing relationship of devices 84 and 88, confirming the results of the previous `symbcv list` command. However, the N/A in place of the BCV logical name and the absence of an asterisk after the BCV device number 88 suggests that the paired BCV does not belong to device group ProdBgrp. (Recall that BCV 89 was added to the device group, not 88.)

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical                Sym     Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001         0084      0 N/A                    0088         0 Split

Total              -------                              -------
  MB(s)                0.0                                  0.0

Legend:

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

- The following attempt to *incrementally establish* or *fully establish* the devices in the device group (84 with 89) will fail because of the current pairing relationship of standard 84 with BCV 88.

```
symmir -g ProdBgrp establish -noprompt

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

Unable to locate the paired BCV device; it may have been disassociated from
the group

symmir -g ProdBgrp establish -noprompt -full

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

Cannot proceed when the device's paired BCV is not associated with the device
group except if the force flag is used
```

- The `-force -full` options would have overridden the current pairing relationship and executed the `establish` operation. Instead, the example shows how the following `symmir attach` command creates a preferred standard/BCV relationship between devices 84 and 89. This attachment will override the standard's prior pairing relationship the next time an establish operation attempts to create the 84/89 pair.

  `symmir -g ProdBgrp attach DEV001 BCV dev 89`

- The `symmir query -attach` command displays both the attachment and the current pairing relationship. The attachment information resides in the SYMAPI database. The pairing information resides in Symmetrix global memory. The absence of an asterisk (*) next to device 88 signifies that 88 is not associated with device group ProdBgrp.

```
symmir -g ProdBgrp -attach query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

Standard Device     Paired BCV Device    Attached BCV Device      State
------------------  -------------------  -------------------    ------------
Logical        Sym  Logical         Sym  Logical         Sym    STD <=> BCV
------------------  -------------------  -------------------    ------------
DEV001        0084  N/A            0088  BCV001         0089 *  Split

Legend:

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

- The `symmir -full establish` command is now successful because establishing an attached pair in the device group overrides any previous pairing relationship these devices might have. An incremental establish operation is not possible because BCV 89 and standard 84 have no prior pairing relationship. The `-v` option generates more detailed output.

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

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]

SELECTING Target devices in the group:

  Device: 0089 [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0089(B) [PAIRED]

STARTING a BCV 'ESTABLISH' operation.

The BCV 'ESTABLISH' operation SUCCEEDED.

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

- The `symmir query` and `symmir query -attach` commands show that the pairing information in the Symmetrix array has been updated to reflect the new BCV pair. The asterisk (*) beside 89 signifies its association with the group.

symmir -g ProdBgrp query

```
Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

     Standard Device                    BCV Device                   State
-------------------------- ------------------------------------- ------------
                    Inv.                                    Inv.
Logical        Sym  Tracks Logical                 Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001         0084      0 BCV001                  0089 *       0 Synchronized

Total              -------                                 -------
  MB(s)                0.0                                     0.0

Legend:

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

symmir -g ProdBgrp -attach query

```
Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

Standard Device      Paired BCV Device    Attached BCV Device      State
------------------   -------------------   -------------------   ------------
Logical        Sym   Logical         Sym    Logical         Sym   STD <=> BCV
------------------   -------------------   -------------------   ------------
DEV001        0084   BCV001         0089 * BCV001         0089 * Synchronized

Legend:

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

- The `symmir detach` command now ends the preferred attachment between standard 84 and BCV 89. The subsequent use of the `symmir query -attach` command confirms the results of the detach action.

symmir -g ProdBgrp detach DEV001 BCV dev 89
symmir -g ProdBgrp query -attach

```
Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

Standard Device      Paired BCV Device    Attached BCV Device      State
------------------   -------------------   -------------------   ------------
Logical        Sym   Logical         Sym    Logical         Sym   STD <=> BCV
------------------   -------------------   -------------------   ------------
DEV001        0084   BCV001         0089 * N/A             N/A   Synchronized

Legend:

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

- Using the `-multi` option with the `symmir query` command causes SYMAPI to interrogate the Symmetrix array about the list of BCVs that can be incrementally established or restored with the standard.
The N/A notation instead of a logical name and the lack of an asterisk (*) after 88 indicates that this BCV is not part of the device group ProdBgrp.

```
symmir -g ProdBgrp -multi query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001          0084       0 BCV001                  0089 *       0 Synchronized
                           0 N/A                     0088         0 Split

Total (Primary)     -------                                -------
  MB(s)                 0.0                                    0.0

Legend:

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

- The `symmir -instant split` command splits the BCV. The `symmir -bg query` command displays the state of background split after the instant split has been initiated. The Split (bg) state indicates that the background split is continuing.

```
symmir -g ProdBgrp -noprompt -instant split; symmir -g ProdBgrp -bg query

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

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

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001          0084       0 BCV001                  0089 *       0 Split    (bg)

Total               -------                                -------
  MB(s)                 0.0                                    0.0

Legend:

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

- The most common use of `symmir verify` is to perform the verify operation with an `-i <seconds>` option, causing a verification to occur every *i* seconds until the BCV pair is established or restored. To demonstrate this feature, it is first necessary to add three more standards and three more BCVs to the device group ProdBgrp. The `symld addall` command adds the standards 85, 86, and 87.

  `symld -g ProdBgrp -sid 3264 addall -RANGE 85:87`

- Removing BCV 89 first from the device group before associating BCVs 88, 89, 8A and 8B allows the example to maintain a logical order. By adding 88, 8A and 8B without first removing 89 would cause 89 to have the logical name BCV001, 88 to have the name BCV002, 8A to have the name BCV003, and 8B to have the name BCV004. By removing 89 before adding the four BCVs together, the logical names will ascend in the same order as the BCV serial numbers. The `symbcv associateall` command associates the four BCVs in the range.

```
symbcv -g ProdBgrp -sid 3264 disassociate dev 89
symbcv -g ProdBgrp -sid 3264 associateall -RANGE 88:8B
symld -g ProdBgrp list

Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

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

DEV001       /dev/dsk/c4t0d0s2 0084 + 16B:0 01A:C0 2-Way Mir          RW     516
DEV002       /dev/dsk/c4t0d1s2 0085   16B:0 02A:C0 2-Way Mir          RW     516
DEV003       /dev/dsk/c4t0d2s2 0086 + 16B:0 01B:C0 2-Way Mir          RW     516
DEV004       /dev/dsk/c4t0d3s2 0087 + 16B:0 02B:C0 2-Way Mir          RW     516

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                     Standard Device          Status
------------------------------------ --------------------------- ------------
                              Inv.                           Inv.
Logical          Sym   RDF Att. Tracks Logical          Sym   Tracks BCV <=> STD
------------------------------------ --------------------------- ------------
BCV001           0088         +       0 DEV001           0084       0 Split
BCV002           0089         +       0 DEV001           0084       0 Split
BCV003           008A         +       0 DEV003           0086       0 Split
BCV004           008B         +       0 DEV004           0087       0 Split

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

Legend for BCV devices:

(+): BCV is paired with a member STD device.
(-): BCV is paired with a non-member STD device.
```

- The `symmir establish -exact` command creates pairs in the exact order that they were added to the device group. Using the `-exact` option overrides any prior pairing relationships that these devices might have.

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

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

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

- The `symmir verify` command with the `-c <count>` option limits the number of times the device group is checked to see if the BCV pairs have finished synchronizing. Without the `-c` option, verify continues to loop until the synchronization is complete.

```
symmir -g ProdBgrp verify -i 60 -c 3

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

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

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

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

symmir -g ProdBgrp verify

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

- Once the BCV pairs are synchronized, the `symmir split` command splits the BCVs.

```
symmir -g ProdBgrp -noprompt split

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

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

- The `symmir restore` command incrementally restores data from the BCVs to the standards.

```
symmir -g ProdBgrp restore -noprompt

'Incremental Restore' operation execution is in progress for device group
'ProdBgrp'. Please wait...

'Incremental Restore' operation successfully initiated for device group
'ProdBgrp'.
symmir -g ProdBgrp verify -i 60

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

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

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

- The following examples demonstrate the use of the special BCV state options with the `symmir verify` command. The following `symmir establish` command uses the `-exact` option again to fully establish the group's BCV pairs.

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

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

'Full Establish' operation successfully initiated for device group 'ProdBgrp'.

symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                      BCV Device                   State
-------------------------- ------------------------------------- ------------
                        Inv.                                 Inv.
Logical          Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001           0084      0 BCV001                   0088 *   15458 SyncInProg
DEV002           0085      0 BCV002                   0089 *   15625 SyncInProg
DEV003           0086      0 BCV003                   008A *   15753 SyncInProg
DEV004           0087      0 BCV004                   008B *   15738 SyncInProg

Total                -------                               -------
  Track(s)                 0                                 62574
  MB(s)                  0.0                                1955.4

Legend:

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

- Before the BCV pairs are synchronized, using the `symmir verify` command without any option returns either the NOT_ALL_SYNCHRONIZED or NONE_SYNCHRONIZED value, whereas using `verify` with the `-synched` option returns the NOT_ALL_SYNCHED or NONE_SYNCHED values. The difference between the SYNCHRONIZED and SYNCHED option is that the former applies in the case of restore and establish operations, while the latter works with establish operations only. Note that the `echo $status` value 5 is the code number for NONE_SYNCHRONIZED. (After the BCV pairs are synchronized, `verify` and `verify -synched` will return a zero, which is the CLI_C_SUCCESS value.)

```
symmir -g ProdBgrp verify

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

echo $status
5

symmir -g ProdBgrp verify -synched

None of the devices in group 'ProdBgrp' are in the 'Synchronized' state.

echo $status
11

symmir -g ProdBgrp verify -syncinprog

All devices in group 'ProdBgrp' are in the 'SyncInProg' state.
```

```
echo $status
0
```

- After some time elapses, the `symmir query` command displays that two of the BCV pairs are fully synchronized and two are still in progress.

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                    BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                   0088 *      0 Synchronized
DEV002          0085      0 BCV002                   0089 *      0 Synchronized
DEV003          0086      0 BCV003                   008A *    414 SyncInProg
DEV004          0087      0 BCV004                   008B *    627 SyncInProg

Total               -------                                -------
  Track(s)                0                                   1041
  MB(s)                 0.0                                   32.5
symmir -g ProdBgrp verify

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

echo $status
4

symmir -g ProdBgrp verify -synched

Not all devices in group 'ProdBgrp' are in the 'Synchronized' state.

echo $status
10

symmir -g ProdBgrp verify -syncinprog

Not all devices in group 'ProdBgrp' are in the 'SyncInProg' state.

echo $status
27
```

- Later all BCV pairs in the group become synchronized, as the `symmir query` command shows.

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type            : REGULAR
DG's Symmetrix ID    : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                   0088 *      0 Synchronized
DEV002          0085      0 BCV002                   0089 *      0 Synchronized
DEV003          0086      0 BCV003                   008A *      0 Synchronized
DEV004          0087      0 BCV004                   008B *      0 Synchronized

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

Legend:

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

- Examine the return codes from the following `symmir verify` commands. Notice that while `verify` and `verify -synched` return the success code 0, `verify -restored` returns a failure code of 13.

```
symmir -g ProdBgrp verify

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

echo $status
0

# symmir -g ProdBgrp -synched verify

All devices in group 'ProdBgrp' are in the 'Synchronized' state.

echo $status
0

symmir -g ProdBgrp -restored verify

None of the devices in group 'ProdBgrp' are in the 'Restored' state.

echo $status
13

symmir -g ProdBgrp -syncinprog verify

None of the devices in group 'ProdBgrp' are in the 'SyncInProg'
state.

echo $status
28
```

- All BCV pairs in the device group are fully synchronized. The following `symmir split` commands split the devices in stages to show the values returned by the `symmir verify -split` commands.

```
symmir -g ProdBgrp -split verify

None of the devices in group 'ProdBgrp' are in the 'Split' state.

echo $status
26

symmir -g ProdBgrp split DEV001 -noprompt

'Split' operation execution is in progress for device 'DEV001' in device group
'ProdBgrp'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'ProdBgrp'.

symmir -g ProdBgrp -split verify

Not all devices in group 'ProdBgrp' are in the 'Split' state.

echo $status
25
```

- The following `symmir split` command splits the rest of the devices to show the values returned by the subsequent `symmir verify -split` commands. Using the `-v` option provides more detailed information on the splitting process.

```
symmir -g ProdBgrp -noprompt split -v

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

SELECTING Standard devices in the group:

  Device: 0084 [SKIPPED - ALRDY IN STATE]
  Device: 0085 [SELECTED]
  Device: 0086 [SELECTED]
  Device: 0087 [SELECTED]

SELECTING Target devices in the group:

  Device: 0088 [SKIPPED]
  Device: 0089 [SELECTED]
  Device: 008A [SELECTED]
  Device: 008B [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0085(S) - 0089(B) [PAIRED]
  Devices: 0086(S) - 008A(B) [PAIRED]
  Devices: 0087(S) - 008B(B) [PAIRED]

STARTING a BCV 'SPLIT' operation.

The BCV 'SPLIT' operation SUCCEEDED.

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

symmir -g ProdBgrp -split verify

All devices in group 'ProdBgrp' are in the 'Split' state.

echo $status
0
```

- The following commands demonstrate the verify return codes in the course of a `restore` operation.

```
symmir -g ProdBgrp -noprompt -full restore

'Full Restore' operation execution is in progress for device group 'ProdBgrp'.
Please wait...

'Full Restore' operation successfully initiated for device group 'ProdBgrp'.
symmir query -g ProdBgrp

Device Group (DG) Name: ProdBgrp
DG's Type            : REGULAR
DG's Symmetrix ID    : 000000003264

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                  Sym     Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084  12305 BCV001                   0088 *       0 RestInProg
DEV002          0085  16500 BCV002                   0089 *       0 RestInProg
DEV003          0086  14164 BCV003                   008A *       0 RestInProg
DEV004          0087  13827 BCV004                   008B *       0 RestInProg

Total               -------                               -------
  Track(s)            56796                                     0
  MB(s)             1774.9                                    0.0

Legend:

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

symmir -g ProdBgrp verify

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

echo $status
5

symmir -g ProdBgrp verify -restored

None of the devices in group 'ProdBgrp' are in the 'Restored' state.

echo $status
13

symmir -g ProdBgrp verify -restinprog

All devices in group 'ProdBgrp' are in the 'RestInProg' state.

echo $status
0
```

- After some time elapses, the `symmir query` command displays that two of the BCV pairs are fully restored and two are still in progress.

```
symmir query -g ProdBgrp

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                 Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                  0088 *      0 Restored
DEV002          0085      0 BCV002                  0089 *      0 Restored
DEV003          0086   2567 BCV003                  008A *      0 RestInProg
DEV004          0087   4468 BCV004                  008B *      0 RestInProg

Total               -------                            -------
  Track(s)             7035                                  0
  MB(s)               219.8                                0.0

Legend:

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

symmir -g ProdBgrp verify

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

echo $status
4

symmir -g ProdBgrp -restored verify

Not all devices in group 'ProdBgrp' are in the 'Restored' state.

echo $status
12

symmir -g ProdBgrp -restinprog verify

Not all devices in group 'ProdBgrp' are in the 'RestInProg' state.

echo $status
29
```

- As the `symmir query` command shows, all devices are now restored.

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                 Inv.
Logical         Sym  Tracks Logical                 Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                  0088 *      0 Restored
DEV002          0085      0 BCV002                  0089 *      0 Restored
DEV003          0086      0 BCV003                  008A *      0 Restored
DEV004          0087      0 BCV004                  008B *      0 Restored

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

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

- Examine the return codes from the following `symmir verify` commands. Notice that while `verify` and `verify -restored` return the success code 0, `verify -synched` returns a failure code of 11.

```
symmir -g ProdBgrp verify

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

echo $status
0

symmir -g ProdBgrp -restored verify

All devices in group 'ProdBgrp' are in the 'Restored' state.

echo $status
0

symmir -g ProdBgrp -synched verify

None of the devices in group 'ProdBgrp' are in the 'Synchronized' state.

echo $status
11

symmir -g ProdBgrp -restinprog verify

None of the devices in group 'ProdBgrp' are in the 'RestInProg' state.
echo $status
30
```

- The `symmir split` command splits all BCV pairs in the device group.

```
symmir -g ProdBgrp split -noprompt

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

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

- The concurrent BCV feature of TimeFinder allows you to establish two BCVs simultaneously with the same standard device. The following `symmir establish` command establishes DEV001 with BCV003 and BCV004. Because these two BCVs do not have a previous pairing relationship with DEV001, the `-full` option is required to copy all of the data from DEV001 to both BCVs.

`symmir -g ProdBgrp -full est DEV001 bcv ld BCV003 DEV001 bcv ld BCV004 -nop`

```
'Full Establish' operation execution is in progress for the device list in
device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for the device list in
device group 'ProdBgrp'.
```

- The `symmir query` command with the `-multi` option displays those standard devices in the group that have a BCV pairing relationship. Now DEV001 is split from BCV001 and is in the process of synchronizing with BCV003 and BCV004. DEV002 is split from BCV002. Because the standards DEV003 and DEV004 no longer have BCV pairing relationships, they are not displayed.

`symmir -g ProdBgrp query -multi`

```
Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                    BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                  Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001         0084      0 BCV003                   008A * 138090 SyncInProg
                         0 BCV004                   008B * 138090 SyncInProg
                         8 BCV001                   0088 *      0 Split
DEV002         0085      0 BCV002                   0089 *      0 Split

Total (Primary)     ------                               -------
  Track(s)               0                                138090
  MB(s)                0.0                                4315.3

Legend:

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

Example 2: Querying a composite group

This example was performed using Solutions Enabler Version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (sid 35 and 60). A composite group defined on the host contains a set of BCV pairs that spans the two Symmetrix arrays. The devices include standard devices and BCV devices. BCV pairs in the composite group are created, activated, and controlled together.

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

```
symcg create TimeFinder -type regular
```

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

```
symcg -cg TimeFinder addall dev -range 56:6E -sid 35
symcg -cg TimeFinder addall dev -range 14:27 -sid 60
```

- The following `symbcv` commands add to the composite group a range of BCV devices from each of the two source Symmetrix arrays.

```
symbcv -cg TimeFinder associateall dev -range 182:19A -sid 35
symbcv -cg TimeFinder associateall dev -range 3B6:3C9 -sid 60
```

- The `symmir establish` command creates optimized standard/BCV pairings among devices within each Symmetrix array and performs a full establish operation on them.

```
symmir -cg TimeFinder establish -full -optimize -noprompt

'Full Establish' operation execution is in progress for composite
group
'TimeFinder'. Please wait...

'Full Establish' operation successfully initiated for composite
group
'TimeFinder'.
```

- The `symmir query` command shows synchronization is in progress (SyncInProg) for the BCV pairs. Note that devices 56 and 59 are meta devices (as are their paired BCV devices), and only the meta head device is displayed.

symmir -cg TimeFinder query

```
Composite Group Name       :  TimeFinder
Composite Group Type       :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID               : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                  State
-------------------------- --------------------------------- ------------
                    Inv.                                Inv.
Sym                 Tracks Sym               Att.       Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0056                     0 0182                 *      344150 SyncInProg
0059                     0 0185                 *      228145 SyncInProg
005B                     0 0187                 *      120030 SyncInProg
005C                     0 0188                 *      120181 SyncInProg
005D                     0 0189                 *      120878 SyncInProg
005E                     0 018A                 *      110078 SyncInProg
005F                     0 018B                 *       41523 SyncInProg
0060                     0 018C                 *       33900 SyncInProg
0061                     0 018D                 *      123432 SyncInProg
0062                     0 018E                 *       17008 SyncInProg
0063                     0 018F                 *       53389 SyncInProg
0064                     0 0190                 *       42266 SyncInProg
0065                     0 0191                 *        7995 SyncInProg
0066                     0 0192                 *       37293 SyncInProg
0067                     0 0193                 *       45200 SyncInProg
0068                     0 0194                 *       44386 SyncInProg
0069                     0 0195                 *       45943 SyncInProg
006A                     0 0196                 *       55380 SyncInProg
006B                     0 0197                 *       55525 SyncInProg
006C                     0 0198                 *       43544 SyncInProg
006D                     0 0199                 *       48643 SyncInProg
006E                     0 019A                 *       13424 SyncInProg

Symmetrix ID               : 000184500160    (Microcode Version: 5568)

      Standard Device                BCV Device                  State
-------------------------- --------------------------------- ------------
                    Inv.                                Inv.
Sym                 Tracks Sym               Att.       Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0014                     0 03B6                 *      131367 SyncInProg
0015                     0 03B7                 *      120967 SyncInProg
0016                     0 03B8                 *      114601 SyncInProg
0017                     0 03B9                 *      101797 SyncInProg
0018                     0 03BA                 *      122833 SyncInProg
0019                     0 03BB                 *      117237 SyncInProg
001A                     0 03BC                 *      115669 SyncInProg
001B                     0 03BD                 *      121046 SyncInProg
001C                     0 03BF                 *      132493 SyncInProg
001D                     0 03BE                 *      125720 SyncInProg
001E                     0 03C0                 *      113095 SyncInProg
001F                     0 03C1                 *      103544 SyncInProg
0020                     0 03C2                 *      124720 SyncInProg
0021                     0 03C3                 *      109486 SyncInProg
0022                     0 03C4                 *      109383 SyncInProg
0023                     0 03C5                 *      122115 SyncInProg
0024                     0 03C6                 *      133037 SyncInProg
0025                     0 03C7                 *      110196 SyncInProg
```

```
0026                     0 03C8                     *      118622 SyncInProg
0027                     0 03C9                     *      118089 SyncInProg

Total               ------                           -------
  Track(s)               0                           4118330
  MB(s)                  0                            128698

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
```

- The `symmir split` command performs an instant split on all BCV pairs in the composite group.

  `symmir -cg TimeFinder split -instant -noprompt`

  ```
  'Split' operation execution is in progress for composite group
  'TimeFinder'.
  Please wait...

  'Split' operation successfully executed for composite group
  'TimeFinder'.
  ```

- The `symmir query` command with the `-bg` and `-percent` options shows the percentage completion of the split that is currently occurring in the background.

`symmir -cg TimeFinder query -bg -percent`

```
Composite Group Name      :  TimeFinder
Composite Group Type      :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID             : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                 State      Split
-------------------------- --------------------------------- ------------ -----
                    Inv.                                Inv.
Sym                 Tracks Sym              Att.       Tracks STD <=> BCV   (%)
-------------------------- --------------------------------- ------------ -----
0056                     0 0182                     *          0 Split    (bg)  14
0059                     0 0185                     *          0 Split    (bg)  14
005B                     0 0187                     *          0 Split    (bg)  15
005C                     0 0188                     *          0 Split    (bg)  15
005D                     0 0189                     *          0 Split    (bg)  15
005E                     0 018A                     *          0 Split    (bg)  13
005F                     0 018B                     *          0 Split    (bg)  15
0060                     0 018C                     *          0 Split    (bg)  15
0061                     0 018D                     *          0 Split    (bg)  15
0062                     0 018E                     *          0 Split    (bg)  13
0063                     0 018F                     *          0 Split    (bg)  15
0064                     0 0190                     *          0 Split    (bg)  15
0065                     0 0191                     *          0 Split    (bg)  15
0066                     0 0192                     *          0 Split    (bg)  13
0067                     0 0193                     *          0 Split    (bg)  15
0068                     0 0194                     *          0 Split    (bg)  15
0069                     0 0195                     *          0 Split    (bg)  15
006A                     0 0196                     *          0 Split    (bg)  13
006B                     0 0197                     *          0 Split    (bg)  15
006C                     0 0198                     *          0 Split    (bg)  15
006D                     0 0199                     *          0 Split    (bg)  15
006E                     0 019A                     *          0 Split    (bg)  13

Symmetrix ID             : 000184500160    (Microcode Version: 5568)
```

```
      Standard Device                   BCV Device                  State      Split
-------------------------- --------------------------------- ------------ -----
                    Inv.                                Inv.
Sym                 Tracks Sym                 Att.       Tracks STD <=> BCV    (%)
-------------------------- --------------------------------- ------------ -----
0014                     0 03B6                  *             0 Split    (bg)    5
0015                     0 03B7                  *             0 Split    (bg)    5
0016                     0 03B8                  *             0 Split    (bg)    5
0017                     0 03B9                  *             0 Split    (bg)    2
0018                     0 03BA                  *             0 Split    (bg)    2
0019                     0 03BB                  *             0 Split    (bg)    2
001A                     0 03BC                  *             0 Split    (bg)    2
001B                     0 03BD                  *             0 Split    (bg)    2
001C                     0 03BF                  *             0 Split    (bg)    5
001D                     0 03BE                  *             0 Split    (bg)    2
001E                     0 03C0                  *             0 Split    (bg)    9
001F                     0 03C1                  *             0 Split    (bg)    8
0020                     0 03C2                  *             0 Split    (bg)    6
0021                     0 03C3                  *             0 Split    (bg)    6
0022                     0 03C4                  *             0 Split    (bg)   18
0023                     0 03C5                  *             0 Split    (bg)   17
0024                     0 03C6                  *             0 Split    (bg)    9
0025                     0 03C7                  *             0 Split    (bg)    8
0026                     0 03C8                  *             0 Split    (bg)    6
0027                     0 03C9                  *             0 Split    (bg)    7

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

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
(bg): The paired BCV device is splitting in the background.
```

- The `symmir verify` command checks the background split at 60-second intervals until the split operation on the composite group has completed.

`symmir -cg TimeFinder verify -bg -split -i 60`

```
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
All devices in Composite Group 'TimeFinder' have finished splitting in the
background.
```

8

Setting Up TimeFinder/Mirror BCV Pairs

This chapter provides examples on setting up BCV pairs in the TimeFinder/Mirror environment. It focuses on creating and populating a device group and creating a BCV pair from a standard and BCV that belong to the same device group.

The following examples illustrate TimeFinder functionality in versions of EMC Solutions Enabler up through Version 6.0 running on Symmetrix arrays using Enginuity Versions up through 5x71.

Examples

The following examples show BCVs with physical device names, signifying that the BCVs are addressable by the host on which the standards are visible. While using a single host is convenient for demonstration purposes, usually a second host is used for accessing BCVs for business continuance tasks such as backups and testing.

- The `symdg` command creates a device group (ProdBgrp). The `symld` commands add standard devices to the group, using either a device's physical device (pd) name or its Symmetrix device (dev) name. The `symbcv` command associates with the device group either one BCV or a range of BCVs as shown here:

```
symdg create ProdBgrp
symld -g ProdBgrp add pd /dev/rdsk/c4t0d0s2
symld -g ProdBgrp -sid 3264 add dev 85
symld -g ProdBgrp add pd /dev/rdsk/c4t0d2s2
symbcv -g ProdBgrp -sid 3264 -RANGE 88:8A associateall dev
symdg show ProdBgrp

Group Name:  ProdBgrp

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

    Number of STD Devices in Group                 :    3
    Number of Associated GK's                      :    0
    Number of Locally-associated VDEV's            :    0
    Number of Locally-associated BCV's             :    3
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (3):
        {

--------------------------------------------------------------------
                                                    Sym              Cap
       LdevName              PdevName                  Dev  Att. Sts
(MB)

--------------------------------------------------------------------
       DEV001              /dev/rdsk/c4t0d0s2      0084      RW
516
       DEV002              /dev/rdsk/c4t0d1s2      0085      RW
516
       DEV003              /dev/rdsk/c4t0d2s2      0086      RW
516
        }
```

- As shown in the LdevName column of the previous display, default logical device names were assigned.

  The following `symmir establish` command matches a BCV pair explicitly, DEV001 with BCV001.

```
symmir -g ProdBgrp -full establish DEV001 BCV ld BCV001 -noprompt

'Full Establish' operation execution is in progress for device 'DEV001'
paired with BCV device 'BCV001' in device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in
group 'ProdBgrp' paired with BCV device 'BCV001'.
```

- The following `symmir verify` command verifies when the BCV pair reaches the Synchronized state. A new message is displayed every 30 seconds until the pair is established. Then the verify loop ends automatically. The time to reach the Synchronized or Restored state varies with the number of devices being established and the amount of data being copied to each BCV. Because only one device in the device group has been established, it is necessary to verify only the progress of that one device as opposed to the whole device group.

```
symmir -g ProdBgrp verify -i 30 DEV001

Device 'DEV001' in group 'ProdBgrp' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'ProdBgrp' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'ProdBgrp' is NOT in the 'Synchronized or Restored'
state.

Device 'DEV001' in group 'ProdBgrp' is in the 'Synchronized or Restored' state.

symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                  0088 *         0 Synchronized

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

Legend:

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

- The following `symmir split` command attempts to split all BCV pairs in the device group.

```
symmir -g ProdBgrp split -noprompt

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

Cannot proceed because the device does not have any BCV pairing relationship
```

- Pairs not previously established cannot be split. Only one BCV pair in the group is currently established. You can use the `-force` option when one or more devices in a group may not be in the expected state for the split operation. As shown below, the `-force` option splits devices that are established, skipping over those that are not established. The `-v` option provides a more detailed output.

```
symmir -g ProdBgrp split -noprompt -force -v

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]
  Device: 0085 [SKIPPED - NOT PAIRED]
  Device: 0086 [SKIPPED - NOT PAIRED]

SELECTING Target devices in the group:

  Device: 0088 [SELECTED]
  Device: 0089 [SKIPPED]
  Device: 008A [SKIPPED]

PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0088(B) [PAIRED]

STARTING a BCV 'SPLIT' operation.

The BCV 'SPLIT' operation SUCCEEDED.

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

- The following `symmir establish` command matches BCV pairs using optimization (`-opt`).

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

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]
  Device: 0085 [SELECTED]
  Device: 0086 [SELECTED]

SELECTING Target devices in the group:

  Device: 0088 [SELECTED]
  Device: 0089 [SELECTED]
  Device: 008A [SELECTED]
```

```
PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0089(B) [PAIRED]
  Devices: 0085(S) - 0088(B) [PAIRED]
  Devices: 0086(S) - 008A(B) [PAIRED]

STARTING a BCV 'ESTABLISH' operation.

The BCV 'ESTABLISH' operation SUCCEEDED.

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

- The following `symmir query` command shows that the establish operation has been initiated, but the pair synchronization is still in progress (SyncInProg). The Invalid Tracks column indicates the number of tracks that are not yet synchronized. Using `-opt` has changed the previously established BCV pairing (84 with 88) to a pairing of 84 with 89.

`symmir -g  ProdBgrp query`

```
Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical          Sym  Tracks Logical               Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001           0084      0 BCV002                0089 *     11196 SyncInProg
DEV002           0085      0 BCV001                0088 *     11551 SyncInProg
DEV003           0086      0 BCV003                008A *     12515 SyncInProg

Total                -------                                -------
  Track(s)                 0                                  35262
  MB(s)                  0.0                                 1101.9

Legend:

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

- The following `symmir verify` command verifies when the BCV pairs reach the Synchronized state. A new message is displayed every 30 seconds, although it may describe the same state as the previous message. The time to reach the Synchronized or Restored state varies with the number of devices being established and the amount of data being copied to each BCV.

`symmir -g ProdBgrp -i 30 verify`

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

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

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

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

- The following `symmir query` command confirms that all BCV pairs in the group are now in the Synchronized state.

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001         0084      0 BCV002                  0089 *         0 Synchronized
DEV002         0085      0 BCV001                  0088 *         0 Synchronized
DEV003         0086      0 BCV003                  008A *         0 Synchronized

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

Legend:

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

symmir -g ProdBgrp split -noprompt

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

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

- The following `symmir establish` command with `-exact` changes the pairings that were set up with optimization.

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

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

'Full Establish' operation successfully initiated for device group 'ProdBgrp'.

symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001         0084      0 BCV001                  0088 *     13484 SyncInProg
DEV002         0085      0 BCV002                  0089 *     14014 SyncInProg
DEV003         0086      0 BCV003                  008A *     14324 SyncInProg

Total               -------                                 -------
  Track(s)                0                                   41822
  MB(s)                 0.0                                  1306.9

Legend:

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

Note: Note that the device pairings have changed as a result of the `-exact` option. The pairings are now in the order that the standards and BCVs were added to the device group.

```
symmir -g ProdBgrp -i 30 verify

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

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

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

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

- This final `symmir query` confirms that the BCV pairs have reached the Synchronized state.

```
symmir -g ProdBgrp query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                    State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                  0088 *         0 Synchronized
DEV002          0085      0 BCV002                  0089 *         0 Synchronized
DEV003          0086      0 BCV003                  008A *         0 Synchronized

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

Legend:

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

- The `symmir split` command splits all BCV pairs in the device group.

```
symmir -g ProdBgrp split -noprompt

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

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

- Although the concurrent BCV feature allows you to establish a standard with two BCVs simultaneously, it is sometimes more efficient to first establish one BCV and then the other. In this case, BCV001 has a current pairing relationship with DEV001, requiring an incremental establish in which only the changes since the split are copied to the BCV. Because BCV002 does not have a current pairing relationship with DEV001, it requires a full establish in which all data from the standard is copied to the BCV. Thus, establishing the two BCVs simultaneously would require a full establish on both BCVs. Instead, as the first step in the two-step method, the following `symmir establish` command re-establishes BCV001 with DEV001 incrementally.

```
symmir -g ProdBgrp establish DEV001 bcv ld BCV001 -noprompt

'Incremental Establish' operation execution is in progress for device
'DEV001' paired with BCV device 'BCV001' in device group 'ProdBgrp'. Please
wait...

'Incremental Establish' operation successfully initiated for device 'DEV001'
in device group 'ProdBgrp' paired with BCV device 'BCV001'.
```

- As the second step to establishing the concurrent BCVs, the following `symmir establish` command fully establishes DEV001 with the second BCV of the concurrent pair. Because these two devices do not have a current pairing relationship with one another, the `-full` option is required to copy all of the data from DEV001 to BCV002. You can perform this second establish operation while DEV001 and BCV001 are still synchronizing (that is, while their state is SyncInProg).

```
symmir -g ProdBgrp -full establish DEV001 bcv ld BCV002 -noprompt

'Full Establish' operation execution is in progress for device 'DEV001'
paired with BCV device 'BCV002' in device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in
device group 'ProdBgrp' paired with BCV device 'BCV002'.
```

- The `symmir query` with the `-multi` option displays those standard devices in the group that have a BCV pairing relationship. Now DEV001 is synchronized with BCV001 and is in the process of synchronizing with BCV002. DEV003 is split from BCV003. Because the standard DEV002 no longer has a BCV pairing relationship, it is not displayed.

`symmir -g ProdBgrp query -multi`

```
Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV002                  0089 *     138090 SyncInProg
                          0 BCV001                  0088 *          0 Synchronized
DEV003          0086      0 BCV003                  008A *          0 Split

Total (Primary)     ------                                  -------
  Track(s)               0                                   138090
  MB(s)                0.0                                   4315.3

Legend:

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

9

Setting Up TimeFinder/Mirror Multiple and Concurrent BCVs

This chapter provides examples on setting up multiple and concurrent BCVs in the TimeFinder/Mirror environment. It focuses on pairing one standard device sequentially with multiple BCVs and pairing one standard device simultaneously with two BCVs.

The following examples illustrate TimeFinder functionality included in versions of EMC Solutions Enabler up to Version 6.0 running on Symmetrix arrays using Enginuity Versions up to 5x71.

Example 1: Pairing one standard sequentially with multiple BCVs

The examples show BCVs with physical device names, signifying that the BCVs are addressable by the host on which the standards are visible. While using a single host is convenient for demonstration purposes, usually a second host is used for accessing BCVs for business continuance tasks.

All splits are instant splits. Including the `-instant` option on the `symmir split` command line causes SYMCLI to return immediately after the foreground split. Omitting the `-instant` option causes SYMCLI to wait until the background split completes before returning.[1]

- Creating a device group and adding devices to it are prerequisites for establishing BCV pairs. The following commands create the device group ProdBgrp and set environment variables to eliminate the need for typing repetitive options.

```
symdg create ProdBgrp
```

- Setting `DG` prevents you from having to type `-g ProdBgrp` in `symld` and `symmir` commands. Setting `SID` identifies the Symmetrix device number for commands such as `symld` and `symbcv` that may require it. Setting `NOPROMPT` disables prompting for control commands. Setting `LDEV_NAMING` assigns physical device numbers as logical names as opposed to `DEV` or `BCV` type names.

```
setenv SYMCLI_DG ProdBgrp
setenv SYMCLI_SID 3264
setenv SYMCLI_NOPROMPT 1
setenv SYMCLI_LDEV_NAMING PDEV
```

- The following SYMCLI command displays currently defined environment variables:

```
symcli -def

Symmetrix Command Line Interface (SYMCLI) Version V5.4-154 (Rev 5.0)
built with SYMAPI Version V5.4-154 (Rev 5.0)

Current settings of the SYMCLI environmental variables:

  SYMCLI_SID              : 3264
  SYMCLI_DG               : ProdBgrp
  SYMCLI_NOPROMPT         : 1
  SYMCLI_LDEV_NAMING      : PDEV
```

- The following `symld` command adds standard device 84 to the device group. With the current environment variable settings, there is no need to include `-g ProdBgrp -sid 3264` in the command.

```
symld add dev 84
```

- The following command adds all ungrouped BCVs in the range 88 to 98 to the device group. When using the `symbcv` command as shown here to add multiple BCVs to a device group, you can substitute the syntax `addall` for `associateall`. When adding a single BCV, you can substitute the syntax `add` for `associate`.

```
symbcv -RANGE 88:98 associateall dev
```

1. Another way to make SYMCLI wait for the background split to complete is to set the SYMAPI_WAIT_FOR_BG_SPLIT variable in the options file located in the SYMAPI configuration directory. Setting this variable to TRUE causes the wait regardless of whether the `symmir split` command includes the `-instant` option.

- The `symdg show` command displays devices in the group. Note in this display that the logical device names match the physical device names. If the LDEV_NAMING environment variable is not set, or if the devices are invisible to the host, the names default to DEV001 for the standard device and BCV001, BCV002, and so forth for the BCVs.

```
symdg show ProdBgrp

Group Name:  ProdBgrp

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

    Number of STD Devices in Group               :    1
    Number of Associated GK's                    :    0
    Number of Locally-associated VDEV's          :    0
    Number of Locally-associated BCV's           :   17
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (1):
        {
        -----------------------------------------------------------------
                                                       Sym               Cap
        LdevName              PdevName                 Dev  Att. Sts     (MB)
        -----------------------------------------------------------------
        c4t0d0s2              /dev/rdsk/c4t0d0s2       0084      RW       516
        }

    BCV Devices Locally-associated (17):
        {
        -----------------------------------------------------------------
                                                       Sym               Cap
        LdevName              PdevName                 Dev  Att. Sts     (MB)
        -----------------------------------------------------------------
        c4t1d0s2              /dev/rdsk/c4t1d0s2       0088      RW       516
        c4t1d1s2              /dev/rdsk/c4t1d1s2       0089      RW       516
        c4t1d2s2              /dev/rdsk/c4t1d2s2       008A      RW       516
        c4t1d3s2              /dev/rdsk/c4t1d3s2       008B      RW       516
        c4t1d4s2              /dev/rdsk/c4t1d4s2       008C      RW       516
        c4t1d5s2              /dev/rdsk/c4t1d5s2       008D      RW       516
        c4t1d6s2              /dev/rdsk/c4t1d6s2       008E      RW       516
        c4t1d7s2              /dev/rdsk/c4t1d7s2       008F      RW       516
        c4t2d0s2              /dev/rdsk/c4t2d0s2       0090      RW       516
        c4t2d1s2              /dev/rdsk/c4t2d1s2       0091      RW       516
        c4t2d2s2              /dev/rdsk/c4t2d2s2       0092      RW       516
        c4t2d3s2              /dev/rdsk/c4t2d3s2       0093      RW       516
        c4t2d4s2              /dev/rdsk/c4t2d4s2       0094      RW       516
        c4t2d5s2              /dev/rdsk/c4t2d5s2       0095      RW       516
        c4t2d6s2              /dev/rdsk/c4t2d6s2       0096      RW       516
        c4t2d7s2              /dev/rdsk/c4t2d7s2       0097      RW       516
        c4t3d0s2              /dev/rdsk/c4t3d0s2       0098      RW       516
        }
```

- The following command establishes the standard device with BCV 88 as the first pairing in a multi-BCV setup.

```
symmir -full establish c4t0d0s2 BCV dev 88

'Full Establish' operation execution is in progress for device
'c4t0d0s2'
paired with BCV device '0088' in device group 'ProdBgrp'. Please
wait...

'Full Establish' operation successfully initiated for device
'c4t0d0s2' in group 'ProdBgrp' paired with BCV device '0088'.
```

- The `symmir query` commands display the progress of the synchronization. Another way to monitor progress is to use the `verify` action, shown later in this chapter. The ellipsis `(...)` indicates where output was omitted.

```
symmir query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- -----------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical               Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------
c4t0d0s2        0084      0 c4t1d0s2              0088 *      5818 SyncInProg

Total               -------                                -------
  MB(s)                 0.0                                  181.8
............................................................................

symmir query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                  State
-------------------------- ------------------------------------- -----------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical               Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- -----------
c4t0d0s2        0084     0  c4t1d0s2              0088 *      5818 Synchronized

Total              -------                                -------
  MB(s)                0.0                                    0.0
............................................................................
```

- The `symmir split` command performs an instant split on the BCV pair but waits for the background split to complete before returning to the caller. The `symmir establish` commands that follow will pair the standard device with seven more BCVs in succession. Wait for an establish operation to complete before attempting a split operation.

```
symmir split

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

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

symmir -full establish c4t0d0s2 BCV dev 89

'Full Establish' operation execution is in progress for device 'c4t0d0s2'
paired with BCV device '0089' in device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for device 'c4t0d0s2'
in group 'ProdBgrp' paired with BCV device '0089'.

symmir split

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

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

- The following interactive UNIX c-shell script establishes the standard device with each of six BCVs in succession, starting with BCV 8A. The `symmir verify` command runs in a loop, checking the status of the BCV pair at 90-second intervals. Upon completion of synchronization, the pair is split and the next BCV is established with the standard.

```
foreach i (8A 8B 8C 8D 8E 8F)
? symmir -full establish c4t0d0s2 BCV dev $i
? symmir verify -i 90
? symmir split
? end
```

- A `symmir query -multi` command displays a list of BCVs with the most recent BCV occupying the top position.

```
symmir query -multi

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------ --------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------ --------
c4t0d0s2       0084       0 c4t1d7s2           008F *           0  Split
                          0 c4t1d6s2           008E *           0  Split
                          0 c4t1d5s2           008D *           0  Split
                          0 c4t1d4s2           008C *           0  Split
                          0 c4t1d3s2           008B *           0  Split
                          0 c4t1d2s2           008A *           0  Split
                          0 c4t1d1s2           0089 *           0  Split
                          0 c4t1d0s2           0088 *           0  Split

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

Legend:

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

symmir -full establish c4t0d0s2 BCV dev 90

'Full Establish' operation execution is in progress for device 'c4t0d0s2'
paired with BCV device '0090' in device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for device 'c4t0d0s2'
in group 'ProdBgrp' paired with BCV device '0090'.
```

- Because the default setting for MAX_BCV_PAIRS is 8 and BCV_PAIR_POLICY defaults to CANCEL_OLDEST, establishing a ninth BCV will cause the oldest BCV (88) to lose its incremental relationship with standard device 84 and drop off the bottom of the list. The `symmir query` command shows that BCV 90 was added to the head of the list.

symmir query -multi

```
Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- --------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
c4t0d0s2        0084       0 c4t2d0s2                0090 *     13488 SyncInProg
                           0 c4t1d7s2                008F *         0 Split
                           0 c4t1d6s2                008E *         0 Split
                           0 c4t1d5s2                008D *         0 Split
                           0 c4t1d4s2                008C *         0 Split
                           0 c4t1d3s2                008B *         0 Split
                           0 c4t1d2s2                008A *         0 Split
                           0 c4t1d1s2                0089 *         0 Split

Total (Primary)      ------                               -------
  Track(s)                0                                 13488
  MB(s)                 0.0                                 421.5

Legend:

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

- BCV 88 is now in the SplitNoInc state, as the `symld list` command below shows.

```
symld list

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

        Standard Device Name          Directors             Device
----------------------------------   ------------- -----------------------
                                                                          Cap
Logical     Physical           Sym    SA :P DA :IT  Config        Att Sts  (MB)
---------------------------------- ------------- ------------------------
DEV001      /dev/rdsk/c4t0d0s2 0084 + 16B:0 01A:C0  2-Way Mir          RW     516

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                     Standard Device          Status
------------------------------------ --------------------------- --------
                              Inv.                         Inv.
Logical          Sym  RDF Att. Tracks Logical          Sym  Tracks BCV <=> STD
------------------------------------ --------------------------- --------
c4t1d0s2         0088         +      0 c4t0d0s2         0084      0 SplitNoInc
c4t1d1s2         0089         +      0 c4t0d0s2         0084      0 Split
c4t1d2s2         008A         +      0 c4t0d0s2         0084      0 Split
c4t1d3s2         008B         +      0 c4t0d0s2         0084      0 Split
c4t1d4s2         008C         +      0 c4t0d0s2         0084      0 Split
c4t1d5s2         008D         +      0 c4t0d0s2         0084      0 Split
c4t1d6s2         008E         +      0 c4t0d0s2         0084      0 Split
c4t1d7s2         008F         +      0 c4t0d0s2         0084      0 Split
c4t2d0s2         0090         +      0 c4t0d0s2         0084      0 Synchronized

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

- The `symmir split` command performs an instant split on the synchronized BCV pair but waits for the background split to complete before returning to the caller.

```
symmir split

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

'Split' operation successfully executed for device group 'ProdBgrp'.
setenv SYMCLI_BCV_PAIR_POLICY CANCEL_NEWEST
symcli -def

Symmetrix Command Line Interface (SYMCLI) Version V5.4-154 (Rev 5.0) built
with SYMAPI Version V5.4-154 (Rev 5.0)

Current settings of the SYMCLI environmental variables:

  SYMCLI_SID            : 3264
  SYMCLI_DG             : ProdBgrp
  SYMCLI_NOPROMPT       : 1
  SYMCLI_LDEV_NAMING    : PDEV
  SYMCLI_BCV_PAIR_POLICY: CANCEL_NEWEST
```

- When BCV 88 was previously canceled, all record of track changes for this BCV was destroyed. To establish this BCV again requires a full establish operation. An incremental establish operation here would fail.

```
symmir -full establish c4t0d0s2 BCV dev 88

'Full Establish' operation execution is in progress for device 'c4t0d0s2'
paired with BCV device '0088' in device group 'ProdBgrp'. Please wait...

'Full Establish' operation successfully initiated for device 'c4t0d0s2'
in group 'ProdBgrp' paired with BCV device '0088'.
```

- Because the BCV_PAIR_POLICY is now set to CANCEL_NEWEST, the `symmir query` command indicates that the most recently established device (BCV 90) dropped off the list and assumed the SplitNoInc state.

symmir query -multi

```
Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

     Standard Device                      BCV Device                     State
-------------------------- ------------------------------------- --------
                    Inv.                                      Inv.
Logical        Sym  Tracks Logical                  Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
c4t0d0s2       0084      0 c4t1d0s2                 0088 *         0 Synchronized
                         0 c4t1d7s2                 008F *         0 Split
                         0 c4t1d6s2                 008E *         0 Split
                         0 c4t1d5s2                 008D *         0 Split
                         0 c4t1d4s2                 008C *         0 Split
                         0 c4t1d3s2                 008B *         0 Split
                         0 c4t1d2s2                 008A *         0 Split
                         0 c4t1d1s2                 0089 *         0 Split

Total (Primary)    -------                                   -------
  MB(s)                0.0                                       0.0

Legend:

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

- The `symmir split` command performs an instant split on the synchronized BCV pair but waits for the background split to complete before returning to the caller.

```
symmir split

'Split' operation execution is in progress fordevice group 'ProdBgrp'.
Please wait...

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

setenv SYMCLI_BCV_PAIR_POLICY DONT_CANCEL
symcli -def

Symmetrix Command Line Interface (SYMCLI) Version V5.4-154 (Rev 5.0) built
with SYMAPI Version V5.4-154 (Rev 5.0)

Current settings of the SYMCLI environmental variables:

  SYMCLI_SID            : 3264
  SYMCLI_DG             : ProdBgrp
  SYMCLI_NOPROMPT       : 1
  SYMCLI_LDEV_NAMING    : PDEV
  SYMCLI_BCV_PAIR_POLICY: DONT_CANCEL

symmir -full establish c4t0d0s2 BCV dev 90

'Full Establish' operation execution is in progress for device 'c4t0d0s2'
paired with BCV device '0090' in device group 'ProdBgrp'. Please wait...

The maximum number of BCV devices that can be incrementally paired with a
standard device has been exceeded
```

- This time the establish operation fails because the policy is DONT_CANCEL. BCV 90 remains in the SplitNoInc state. To explicitly cancel a BCV pair relationship, use the `symmir cancel` command. The BCV pair must be in the Split state for the cancel operation to work. BCV 89 is in the Split state.

```
symmir cancel c4t0d0s2 BCV dev 89

'Cancel' operation execution is in progress for device 'c4t0d0s2'paired
with BCV device '0089' in device group 'ProdBgrp'. Please wait...

'Cancel' operation successfully executed for device 'c4t0d0s2'in group
'ProdBgrp' paired with BCV device '0089'.
```

- BCV 089 is now in the SplitNoInc state, as the `symld list` command displays.

```
symld list

Device Group (DG) Name: ProdBgrp
DG's Type            : REGULAR
DG's Symmetrix ID    : 000000003264

        Standard Device Name          Directors               Device
----------------------------------   ------------- -----------------------
                                                                     Cap
Logical     Physical           Sym    SA :P DA :IT  Config Att Sts   (MB)
----------------------------------   ------------- -----------------------
DEV001     /dev/rdsk/c4t0d0s2 0084 +  16B:0 01A:C0  2-Way Mir  RW     516

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

             BCV Device                     Standard Device          Status
------------------------------------ --------------------------- --------
                             Inv.                           Inv.
Logical          Sym  RDF Att. Tracks Logical          Sym  Tracks BCV <=> STD
------------------------------------ --------------------------- ----------
c4t1d0s2         0088        +      0 c4t0d0s2         0084       0 Split
c4t1d1s2         0089        +      0 c4t0d0s2         0084       0 SplitNoInc
c4t1d2s2         008A        +      0 c4t0d0s2         0084       0 Split
c4t1d3s2         008B        +      0 c4t0d0s2         0084       0 Split
c4t1d4s2         008C        +      0 c4t0d0s2         0084       0 Split
c4t1d5s2         008D        +      0 c4t0d0s2         0084       0 Split
c4t1d6s2         008E        +      0 c4t0d0s2         0084       0 Split
c4t1d7s2         008F        +      0 c4t0d0s2         0084       0 Split
c4t2d0s2         0090        +      0 c4t0d0s2         0084       0 SplitNoInc
c4t2d1s2         0091               0 N/A               N/A       0 NeverEstab
c4t2d2s2         0092               0 N/A               N/A       0 NeverEstab
c4t2d3s2         0093               0 N/A               N/A       0 NeverEstab
c4t2d4s2         0094               0 N/A               N/A       0 NeverEstab
c4t2d5s2         0095               0 N/A               N/A       0 NeverEstab
c4t2d6s2         0096               0 N/A               N/A       0 NeverEstab
c4t2d7s2         0097               0 N/A               N/A       0 NeverEstab
c4t3d0s2         0098               0 N/A               N/A       0 NeverEstab

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

Legend for BCV devices:

(+): BCV is paired with a member STD device.
(-): BCV is paired with a non-member STD device.
```

- The following interactive UNIX c-shell script initiates incremental establish operations with the BCVs that currently have a pairing relationship with the standard device. If the incremental establish operation returns an error status because the BCV and the standard do not have a valid incremental pairing relationship, the script then initiates a full establish operation.

```
setenv SYMCLI_MAX_BCV_PAIRS 16
foreach i (88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97)
? symmir establish c4t0d0s2 BCV dev $i
? if ($status != 0) symmir establish -full c4t0d0s2 BCV dev $i
? symmir verify -i 30
? symmir split
? end
```

- The following `symmir query` command displays sixteen BCVs that can be incrementally established.

```
symmir query -multi

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                  BCV Device                    State
-------------------------- ------------------------------------- --------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical                Sym     Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
c4t0d0s2       0084      0 c4t2d7s2               0097 *       0 Split
                         0 c4t2d6s2               0096 *       0 Split
                         0 c4t2d5s2               0095 *       0 Split
                         0 c4t2d4s2               0094 *       0 Split
                         0 c4t2d3s2               0093 *       0 Split
                         0 c4t2d2s2               0092 *       0 Split
                         0 c4t2d1s2               0091 *       0 Split
                         0 c4t2d0s2               0090 *       0 Split
                         0 c4t1d7s2               008F *       0 Split
                         0 c4t1d6s2               008E *       0 Split
                         0 c4t1d5s2               008D *       0 Split
                         0 c4t1d4s2               008C *       0 Split
                         0 c4t1d3s2               008B *       0 Split
                         0 c4t1d2s2               008A *       0 Split
                         0 c4t1d1s2               0089 *       0 Split
                         0 c4t1d0s2               0088 *       0 Split

Total (Primary)    -------                              -------
  MB(s)                0.0                                  0.0

Legend:

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

- The `symmir establish` command pairs the most recently established BCV (97) with the standard device if no prior pairing preferences were set up using the attach action. The `-v` option provides a more detailed output.

```
symmir establish -v

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]

SELECTING Target devices in the group:

  Device: 0088 [SELECTED]
  Device: 0089 [SELECTED]
  Device: 008A [SELECTED]
  Device: 008B [SELECTED]
  Device: 008C [SELECTED]
  Device: 008D [SELECTED]
  Device: 008E [SELECTED]
  Device: 008F [SELECTED]
  Device: 0090 [SELECTED]
  Device: 0091 [SELECTED]
  Device: 0092 [SELECTED]
  Device: 0093 [SELECTED]
  Device: 0094 [SELECTED]
  Device: 0095 [SELECTED]
  Device: 0096 [SELECTED]
  Device: 0097 [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0097(B) [PAIRED]

STARTING a BCV 'INCREMENTAL_ESTABLISH' operation.

The BCV 'INCREMENTAL_ESTABLISH' operation SUCCEEDED.

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

- The `symmir query` command confirms that the standard is now paired with BCV c4t2d7s2 (97), its most recent partner.

```
symmir query

Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

     Standard Device                    BCV Device                 State
-------------------------- ------------------------------------- --------
                   Inv.                                 Inv.
Logical         Sym  Tracks Logical               Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------ --------
c4t0d0s2        0084      0 c4t2d7s2             0097 *          0 Synchronized

Total                -------                                -------
  MB(s)                  0.0                                    0.0

Legend:

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

symmir split

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

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

- Using the `symmir attach` command sets up a preferred pairing relationship between the standard device and BCV 90. The subsequent `symmir query` command with the `-attach` option displays the attachment and also that standard device 84 has a current pairing relationship (Split state) with BCV 97.

```
symmir attach c4t0d0s2 BCV dev 90
symmir query -attach

Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

Standard Device      Paired BCV Device     Attached BCV Device      State
------------------   -------------------   -------------------   ----------
Logical        Sym   Logical         Sym   Logical         Sym   STD <=> BCV
------------------   -------------------   -------------------   ----------
c4t0d0s2      0084   c4t2d7s2       0097 * c4t2d0s2       0090 * Split

Legend:

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

- Because of their preferred pairing relationship, the `symmir establish` command pairs the standard device with BCV 90.

```
symmir establish -v

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]

SELECTING Target devices in the group:

  Device: 0088 [SELECTED]
  Device: 0089 [SELECTED]
  Device: 008A [SELECTED]
  Device: 008B [SELECTED]
  Device: 008C [SELECTED]
  Device: 008D [SELECTED]
  Device: 008E [SELECTED]
  Device: 008F [SELECTED]
  Device: 0090 [SELECTED]
  Device: 0091 [SELECTED]
  Device: 0092 [SELECTED]
  Device: 0093 [SELECTED]
  Device: 0094 [SELECTED]
  Device: 0095 [SELECTED]
  Device: 0096 [SELECTED]
  Device: 0097 [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0090(B) [PAIRED]

STARTING a BCV 'INCREMENTAL_ESTABLISH' operation.

The BCV 'INCREMENTAL_ESTABLISH' operation SUCCEEDED.

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

- The `symmir query` command displays the results in the standard format and that the BCV pair is now in the Synchronized state.

```
symmir query

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                   BCV Device                   State
-------------------------- ------------------------------------- --------
                    Inv.                                    Inv.
Logical         Sym  Tracks Logical                 Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
c4t0d0s2        0084      0 c4t2d0s2             0090 *          0 Synchronized

Total                -------                                  -------
  MB(s)                  0.0                                      0.0

Legend:

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

- The `symmir query -attach` command displays a format showing that BCV 90 is the attached BCV and that BCV 90 is the device now having the current pairing relationship (Synchronized state) with standard device 84.

```
symmir query -attach

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

Standard Device      Paired BCV Device    Attached BCV Device      State
------------------   -------------------   -------------------   ----------
Logical       Sym    Logical        Sym     Logical        Sym   STD <=> BCV
------------------   -------------------   -------------------   -----------
c4t0d0s2      0084   c4t2d0s2       0090 * c4t2d0s2       0090 * Synchronized

Legend:

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

Example 2: Concurrent BCVs

The following commands that illustrate concurrent BCVs were issued in a separate SYMCLI session on a different host and Symmetrix array than the host/Symmetrix array used in Example 1. Therefore, the environment variables defined in Example 1 are no longer valid for the new environment.

The following `symdg` command creates a new device group called conbcv. The `symld` commands add two standard devices to the group. The `symbcv` command associates four BCV devices with the group.

```
symdg create conbcv
symld -g conbcv -sid 184502237 add dev 355
symld -g conbcv -sid 184502237 add dev 356
symbcv -g conbcv -sid 184502237 -RANGE 35F:362 associateall dev
```

- The `symdg show` command displays the group's devices and their assigned logical names.

```
symdg show conbcv

Group Name:  conbcv

    Group Type                                   : REGULAR
    Device Group in GNS                          : Yes
    Valid                                        : Yes
    Symmetrix ID                                 : 000184502237
    Group Creation Time                          : Tue Jan  13 11:41:32 2004
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI
    Number of STD Devices in Group               :    2
    Number of Associated GK's                    :    0
    Number of Locally-associated VDEV's          :    0
    Number of Locally-associated BCV's           :    4
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (2):
        {
        ----------------------------------------------------------------
                                                      Sym               Cap
        LdevName              PdevName                Dev  Att. Sts     (MB)
        -----------------------------------------------------------------
        DEV001                /dev/rdsk/c4t0d0        0355      RW      4315
        DEV002                /dev/rdsk/c4t0d1        0356      RW      4315
        }

    BCV Devices Locally-associated (4):
        {
        -----------------------------------------------------------------
                                                      Sym               Cap
        LdevName              PdevName                Dev  Att. Sts     (MB)
        -----------------------------------------------------------------
        BCV001                N/A                     035F      RW      4315
        BCV002                N/A                     0360      RW      4315
        BCV003                N/A                     0361      RW      4315
        BCV004                N/A                     0362      RW      4315
        }
```

- The `symmir attach` commands create preferred standard/BCV relationships between devices DEV001 and BCV001, and between DEV002 and BCV003.

```
symmir -g conbcv attach DEV001  bcv ld BCV001 -noprompt
symmir -g conbcv attach DEV002  bcv ld BCV003 -noprompt
```

- The following `symmir establish` command creates concurrent BCV pairs. BCV001 and BCV002 are matched with standard DEV001, while BCV003 and BCV004 are matched with DEV002.

```
symmir -g conbcv -full establish DEV001 bcv ld BCV001 DEV001 bcv ld BCV002
                         DEV002 bcv ld BCV003 DEV002 bcv ld BCV004 -noprompt

'Full Establish' operation execution is in progress for the device list in
device group 'conbcv'. Please wait...

'Full Establish' operation successfully initiated for the device list in
device group 'conbcv'.
```

- The `symmir query` with the `-multi` option displays the status of the concurrent BCVs (state is SyncInProg).

```
symmir -g conbcv query -multi

Device Group (DG) Name: conbcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000184502237

     Standard Device                  BCV Device                  State
-------------------------- ----------------------------------- --------
                    Inv.                                 Inv.
Logical        Sym  Tracks Logical              Sym       Tracks STD <=> BCV
-------------------------- ----------------------------------- --------
DEV001         0355      0 BCV001               035F *    127806 SyncInProg
                         0 BCV002               0360 *    127687 SyncInProg
DEV002         0356      0 BCV003               0361 *    130115 SyncInProg
                         0 BCV004               0362 *    130084 SyncInProg

Total (Primary)     ------                             -------
  Track(s)               0                               257921
  MB(s)                0.0                               8060.0

Legend:

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

- The `symmir verify` command displays a message every five seconds until the BCV pairs are in a Synchronized state.

```
symmir -g conbcv verify -synched -concurrent -i 5

None of the devices in group 'conbcv' are in the 'Synchronized'
state.

None of the devices in group 'conbcv' are in the 'Synchronized'
state.

Not all devices in group 'conbcv' are in the 'Synchronized' state.

Not all devices in group 'conbcv' are in the 'Synchronized' state.

All devices in group 'conbcv' are in the 'Synchronized' state.
```

- The `symmir query -multi` command confirms that the concurrent BCV pairs are in the Synchronized state.

```
symmir -g conbcv query -multi

Device Group (DG) Name: conbcv
DG's Type            : REGULAR
DG's Symmetrix ID    : 000184502237

     Standard Device                  BCV Device                 State
-------------------------- ------------------------------------- --------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical                Sym      Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
DEV001         0355      0 BCV001                 035F *        0 Synchronized
                         0 BCV002                 0360 *        0 Synchronized
DEV002         0356      0 BCV003                 0361 *        0 Synchronized
                         0 BCV004                 0362 *        0 Synchronized

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

Legend:

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

- The `symmir split` command performs an instant split operation on all BCV pairs in the device group. Using the `-instant` option causes SYMCLI to return immediately after the foreground split, while the background split is still in progress.

```
symmir -g conbcv split -instant -noprompt

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

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

- The `symmir query` command displays the concurrent BCVs and the status of the background split (`-bg`). The Split (bg) state means that the background split is still in progress.

```
symmir -g conbcv query -multi -bg

Device Group (DG) Name: conbcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000184502237

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- --------
                    Inv.                                    Inv.
Logical         Sym  Tracks Logical                 Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- --------
DEV001          0355      0 BCV002                  0360 *         0 Split   (bg)
                          0 BCV001                  035F *         0 Split   (bg)
DEV002          0356      0 BCV004                  0362 *         0 Split   (bg)
                          0 BCV003                  0361 *         0 Split   (bg)

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

Legend:

(*): The paired BCV device is associated with this group.
(bg): The paired BCV device is splitting in the background.
```

- The `symmir verify` command checks every five seconds until the background split is complete.

```
symmir -g conbcv verify -split -concurrent -i 5 -bg

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

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

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

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

- The following command incrementally re-establishes the preferred BCV pairs that were created earlier with the `attach` action.

```
symmir -g conbcv establish -noprompt -v

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

SELECTING Standard devices in the group:

  Device: 0355 [SELECTED]
  Device: 0356 [SELECTED]
SELECTING BCV devices associated with the group:

  Device: 035F [SELECTED]
  Device: 0360 [SELECTED]
  Device: 0361 [SELECTED]
  Device: 0362 [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0355(S) - 035F(B) [PAIRED]
  Devices: 0356(S) - 0361(B) [PAIRED]

STARTING a BCV 'INCREMENTAL_ESTABLISH' operation.

The BCV 'INCREMENTAL_ESTABLISH' operation SUCCEEDED.

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

- The following query displays the status of the concurrent BCVs and that the attached pairs have been re-established (their state is Synchronized). Those BCVs that were not participants in the previous establish operation are still Split.

```
symmir -g conbcv query -multi

Device Group (DG) Name: conbcv
DG's Type            : REGULAR
DG's Symmetrix ID    : 000184502237

     Standard Device                  BCV Device                    State
-------------------------- ------------------------------------- ----------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical              Sym        Tracks STD <=> BCV
-------------------------- ------------------------------------- ----------
DEV001         0355      0 BCV001               035F *          0 Synchronized
                         0 BCV002               0360 *          0 Split
DEV002         0356      0 BCV003               0361 *          0 Synchronized
                         0 BCV004               0362 *          0 Split

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

Legend:

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

- The following command incrementally re-establishes the remaining concurrent BCVs.

```
symmir -g conbcv establish DEV001 bcv ld BCV002 DEV002 bcv ld BCV004 -noprompt

'Incremental Establish' operation execution is in progress for the device list in
device group 'conbcv'. Please wait...

'Incremental Establish' operation successfully initiated for the device list in
device group 'conbcv'.
```

- Querying the device group displays the status of the concurrent BCVs. All are now in a Synchronized state.

```
symmir -g conbcv query -multi

Device Group (DG) Name: conbcv
DG's Type             : REGULAR
DG's Symmetrix ID     : 000184502237

     Standard Device                  BCV Device                     State
-------------------------- ------------------------------------- ----------
                    Inv.                                   Inv.
Logical        Sym  Tracks Logical              Sym        Tracks STD <=> BCV
-------------------------- ------------------------------------- ----------
DEV001         0355      0 BCV002               0360 *          0 Synchronized
                         0 BCV001               035F *          0 Synchronized
DEV002         0356      0 BCV004               0362 *          0 Synchronized
                         0 BCV003               0361 *          0 Synchronized

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

Legend:

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

10

Performing TimeFinder/Mirror Control Operations

This chapter provides examples on creating and splitting a BCV pair, retrieving original BCV data from a fixed BCV mirror, and performing a BCV action while making the BCV or the standard device temporarily inaccessible to the host.

The following examples illustrate TimeFinder functionality in versions of EMC Solutions Enabler up to Version 6.0 running on Symmetrix arrays using Enginuity Versions up to 5x71.

Example 1: Split operations

The examples below show BCVs with physical device names, signifying that the BCVs are addressable by the host on which the standards are visible. While using a single host is convenient for demonstration purposes, usually a second host is used for accessing BCVs for business continuance tasks such as backups.

- Creating a device group and adding devices to it are prerequisites for establishing BCV pairs. The `symdg` command creates a device group (ProdBgrp). The `symld` commands add standard devices to the group, using a device's Symmetrix device (dev) name. The `symbcv` command associates one or all BCVs with the device group; the `-range` option can be used with the `associateall` action to limit the selection to the BCVs that are within the specified range.

```
symdg create ProdBgrp
symld -g ProdBgrp -sid 3264 add dev 84
symld -g ProdBgrp -sid 3264 add dev 85
symld -g ProdBgrp -sid 3264 add dev 86
symld -g ProdBgrp -sid 3264 add dev 87
symbcv -g ProdBgrp -sid 3264 -RANGE 88:8B associateall dev
symdg show ProdBgrp

Group Name:  ProdBgrp

    Group Type                                   : REGULAR
    Device Group in GNS                          : Yes
    Valid                                        : Yes
    Symmetrix ID                                 : 000000003264
    Group Creation Time                          : Tue Jan  6 11:17:08 2004
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :    4
    Number of Associated GK's                    :    0
    Number of Locally-associated VDEV's          :    0
    Number of Locally-associated BCV's           :    4
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (4):
        {
        --------------------------------------------------------------------
                                                     Sym               Cap
        LdevName              PdevName               Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c4t0d0s2     0084      RW       516
        DEV002                /dev/rdsk/c4t0d1s2     0085      RW       516
        DEV003                /dev/rdsk/c4t0d2s2     0086      RW       516
        DEV004                /dev/rdsk/c4t0d3s2     0087      RW       516
        }

    BCV Devices Locally-associated (4):
        {
        --------------------------------------------------------------------
                                                     Sym               Cap
        LdevName              PdevName               Dev  Att. Sts     (MB)
        --------------------------------------------------------------------
        BCV001                /dev/rdsk/c4t1d0s2     0088      RW       516
        BCV002                /dev/rdsk/c4t1d1s2     0089      RW       516
        BCV003                /dev/rdsk/c4t1d2s2     008A      RW       516
        BCV004                /dev/rdsk/c4t1d3s2     008B      RW       516
        }
```

- The `symmir establish` command matches BCV pairs using the `-exact` option.

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

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

SELECTING Standard devices in the group:

  Device: 0084 [SELECTED]
  Device: 0085 [SELECTED]
  Device: 0086 [SELECTED]
  Device: 0087 [SELECTED]

SELECTING Target devices in the group:

  Device: 0088 [SELECTED]
  Device: 0089 [SELECTED]
  Device: 008A [SELECTED]
  Device: 008B [SELECTED]

PAIRING of Standard and BCV devices:

  Devices: 0084(S) - 0088(B) [PAIRED]
  Devices: 0085(S) - 0089(B) [PAIRED]
  Devices: 0086(S) - 008A(B) [PAIRED]
  Devices: 0086(S) - 008B(B) [PAIRED]

STARTING a BCV 'ESTABLISH' operation.

The BCV 'ESTABLISH' operation SUCCEEDED.

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

- The `symmir verify` command verifies when the BCV pairs reach the Synchronized state. A new message is displayed every 30 seconds until the pair is established. Then the verify loop ends automatically. The time to reach the Synchronized or Restored state varies with the number of devices being established and the amount of data being copied to each BCV.

```
symmir -g ProdBgrp -i 30 verify

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

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

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

- The following command performs an instant split on BCV pairs in a device group ProdBgrp while the group is being subjected to a high write load (70 percent write, 30 percent read). In UNIX, the `timex` command measures the time it takes to execute a command. The *real* time is the elapsed time. The foreground split completed in 0.90 seconds.

```
timex symmir -g ProdBgrp split -noprompt -instant

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

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

real        0.90
user        0.22
sys         0.05
```

- The `symmir verify` command with the `-split` option confirms that the foreground split is complete.

```
symmir -g ProdBgrp verify -split

All devices in group 'ProdBgrp' are in the 'Split' state.
```

- The following `symmir query` command uses the `-bg` option to display the status of the background split. The background split is still continuing, indicated by a state of Split (bg).

```
symmir -g ProdBgrp -bg query

Device Group (DG) Name: ProdBgrp
DG's Type              : REGULAR
DG's Symmetrix ID      : 000000003264

     Standard Device                   BCV Device                    State
-------------------------- ----------------------------------- ----------
                    Inv.                                  Inv.
Logical        Sym  Tracks Logical                Sym       Tracks STD <=> BCV
-------------------------- ----------------------------------- ----------
DEV001         0084      0 BCV001                 0088 *         0 Split   (bg)
DEV002         0085      0 BCV002                 0089 *         0 Split   (bg)
DEV003         0086      0 BCV003                 008A *         0 Split   (bg)
DEV004         0087      0 BCV004                 008B *         0 Split   (bg)

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

Legend:

(*): The paired BCV device is associated with this group.
(bg): The paired BCV device is splitting in the background.
```

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

```
symmir -g ProdBgrp verify -split -bg -i 10

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

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

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

- The `symmir query -bg` command confirms that the background split is complete (state is Split).

```
symmir -g ProdBgrp query -bg

Device Group (DG) Name: ProdBgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003264

     Standard Device                    BCV Device                 State
-------------------------- ----------------------------------- ------------
                    Inv.                                  Inv.
Logical         Sym  Tracks Logical               Sym       Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------
DEV001          0084      0 BCV001                0088 *          0 Split
DEV002          0085      0 BCV002                0089 *          0 Split
DEV003          0086      0 BCV003                008A *          0 Split
DEV004          0087      0 BCV004                008B *          0 Split

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


Legend:

(*): The paired BCV device is associated with this group.
(bg): The paired BCV device is splitting in the background.
```

- To demonstrate the effect of a reverse split on a BCV that is configured to have two mirrors, the following commands temporarily create a new group called ProdAgrp and move one standard and one BCV to this device group. This action takes both devices out of ProdBgrp and places them in ProdAgrp.

```
symdg create ProdAgrp -type regular
symld -g ProdBgrp move DEV001 ProdAgrp
symbcv -g ProdBgrp move ld BCV001 ProdAgrp
symdg show ProdAgrp

Group Name:  ProdAgrp

    Group Type                                   : REGULAR
    Device Group in GNS                          : Yes
    Valid                                        : Yes
    Symmetrix ID                                 : 000000003264
    Group Creation Time                          : Tue Jan  6 18:19:05 2004
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :    1
    Number of Associated GK's                    :    0
    Number of Locally-associated VDEV's          :    0
    Number of Locally-associated BCV's           :    1
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0

    Standard (STD) Devices (1):
        {
        --------------------------------------------------------------------
                                                   Sym                Cap
        LdevName              PdevName             Dev  Att. Sts      (MB)
        --------------------------------------------------------------------
        DEV001                /dev/rdsk/c4t0d0s2   0084       RW       516
        }

    BCV Devices Locally-associated (1):
        {
        --------------------------------------------------------------------
                                                   Sym                Cap
        LdevName              PdevName             Dev  Att. Sts      (MB)
        --------------------------------------------------------------------
        BCV001                /dev/rdsk/c4t1d0s2   0088       RW       516
        }
```

- Because Symmetrix devices 84 and 88 have a previous pairing relationship, it is possible to incrementally establish these devices with the `symmir establish` command.

```
symmir -g ProdAgrp -noprompt establish

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

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

- The `newfs` command creates a file system on the standard volume /dev/rdsk/c4t0d0s2.

```
newfs /dev/rdsk/c4t0d0s2
newfs: /dev/rdsk/c4t0d0s2 last mounted as /mnt
newfs: construct a new file system /dev/rdsk/c4t0d0s2: (y/n)? y
/dev/rdsk/c4t0d0s2:     1054080 sectors in 1098 cylinders of 15 tracks, 64 sectors
        514.7MB in 69 cyl groups (16 c/g, 7.50MB/g, 3584 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 15456, 30880, 46304, 61728, 77152, 92576, 108000, 123424, 138848, 154272,
 169696, 185120, 200544, 215968, 231392, 245792, 261216, 276640, 292064,
 307488, 322912, 338336, 353760, 369184, 384608, 400032, 415456, 430880,
 446304, 461728, 477152, 491552, 506976, 522400, 537824, 553248, 568672,
 584096, 599520, 614944, 630368, 645792, 661216, 676640, 692064, 707488,
 722912, 737312, 752736, 768160, 783584, 799008, 814432, 829856, 845280,
 860704, 876128, 891552, 906976, 922400, 937824, 953248, 968672, 983072,
 998496, 1013920, 1029344, 1044768,

mkdir /stdmnt
```

- The `mount` command mounts the standard device to an existing mount point.

```
mount /dev/dsk/c4t0d0s2 /stdmnt
```

- The `cp` command copies a large file named bigfile to the BCV pair.

```
cp /mnt/bigfile /stdmnt

ls -l /stdmnt
total 989984
-rw-r--r--   1 root     other    506601472 Mar 13 18:21 bigfile
drwx------   2 root     root        8192 Mar 13 17:41 lost+found

umount /stdmnt
```

- The `symmir split` command performs an instant split on the BCV pair but waits for the background split to complete before returning to the caller. If you include the `-instant` option, SYMCLI returns immediately after the foreground split, allowing you to perform other TimeFinder operations while the BCV pair is splitting in the background.

```
symmir -g ProdAgrp -noprompt split -instant

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

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

- The `symdev show` command on the BCV displays the number of tracks that have to be copied from the BCV's moving mirror (Mirror 1) to its fixed mirror (Mirror 2). Because the tracks on Mirror 2 have to be updated, the invalid tracks show up in the second mirror position displayed at the line `Mirror Set Inv. Tracks`. The following output is an abbreviated display.

```
symdev show 88

Symmetrix ID: 000000003264

    Device Physical Name     : /dev/rdsk/c4t1d0s2

    Device Symmetrix Name    : 0088
    Device Serial ID         : 64088320
    Symmetrix ID             : 000000003264

    Device Group Name        : ProdAgrp
    Device Logical Name      : BCV001
..........................................................................
.......

    Mirror Set Type          : [Data,Data,N/A,N/A]

    Mirror Set DA Status     : [RW,RW,N/A,N/A]

    Mirror Set Inv. Tracks   : [0,16470,0,0]
```

- The `symmir verify -bcv_mirrors` command verifies the progress of the synchronization of the BCV's moving mirror with its fixed mirror. The ellipsis (……) represents several lines of repetitive output that were omitted.

```
symmir -g ProdAgrp -bcv_mirrors verify -i 30

None of the devices in group 'ProdAgrp' have a paired BCV device that has its mirrors
in the 'Synchronized' state.
..........................................................................
All devices in group 'ProdAgrp' have a paired BCV device that has its mirrors in
the 'Synchronized' state.

mkdir /bcvmnt
```

- The `mount` command mounts the BCV. The directory listing confirms that the file named bigfile is on the BCV.

```
mount /dev/dsk/c4t1d0s2 /bcvmnt

ls -l /bcvmnt
total 989984
-rw-r--r--   1 root     other    506601472 Mar 13 18:21 bigfile
drwx------   2 root     root          8192 Mar 13 17:41 lost+found

umount /bcvmnt
```

- The following `symmir establish` command re-establishes the BCV with the standard. The `-reverse` option ensures that the BCV's two mirrors are synchronized, qualifying them for a successful reverse split operation later.

```
symmir -g ProdAgrp -noprompt -reverse establish

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

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

- The following commands replace file bigfile on the BCV pair with a new file named bigfile2.

```
mount /dev/dsk/c4t0d0s2 /stdmnt

ls /stdmnt
bigfile     lost+found

rm /stdmnt/bigfile

cp /mnt/bigfile2 /stdmnt

ls -l /stdmnt
total 884368
-rw-r--r--   1 root     other    452551168 Mar 13 19:09 bigfile2
drwx------   2 root     root          8192 Mar 13 17:41 lost+found

umount /stdmnt
```

- The following command initiates a reverse split operation in which original data (bigfile) is copied from the BCV's fixed mirror to its moving mirror, replacing the more recent data (bigfile2).

```
symmir -g ProdAgrp -noprompt -reverse split

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

'Reverse Split' operation successfully executed for device group 'ProdAgrp'.
```

- While the BCV mirrors are synchronizing, a `symdev show` command on the BCV displays that the moving mirror (Mirror 1) has invalid tracks because its tracks are being updated from the fixed mirror (Mirror 2).

```
symdev show 88

Symmetrix ID: 000000003264

    Device Physical Name      : /dev/rdsk/c4t1d0s2

    Device Symmetrix Name     : 0088
    Device Serial ID          : 64088320
    Symmetrix ID              : 000000003264

    Device Group Name         : ProdAgrp
    Device Logical Name       : BCV001
.........................................................................
    Mirror Set Type           : [Data,Data,N/A,N/A]

    Mirror Set DA Status      : [RW,RW,N/A,N/A]

    Mirror Set Inv. Tracks    : [13860,0,0,0]
```

- While the moving mirror is being refreshed from the fixed mirror, any attempt to restore data from the BCV to the standard will fail (as the following `symmir restore` command demonstrates).

```
symmir -g ProdAgrp restore -noprompt

'Incremental Restore' operation execution is in progress for device group
'ProdAgrp'. Please wait...

BCV Library error is: The operation cannot be executed because the moving
mirror of the BCV is invalid
```

- However, if `symmir restore` is issued with the `-reverse` option and the `-i 30` option, TimeFinder delays the restore operation until both BCV mirrors are synchronized. An alternative to this approach is to use the verify `-bcv_mirrors` option demonstrated earlier. The ellipsis (……) represents several lines of repetitive output that were omitted.

```
symmir -g ProdAgrp restore -noprompt -reverse -i 30

'Incremental Restore' operation execution is in progress for device group
'ProdAgrp'. Please wait...

Performing this operation at this time will not allow you to perform the
next BCV split as a reverse split
.........................................................................
'Incremental Restore' operation execution is in progress for device group
'ProdAgrp'. Please wait...

'Incremental Restore' operation successfully initiated for device group
'ProdAgrp'.
```

- As soon as the restore operation is initiated, data on the BCV becomes accessible to the standard. The following commands mount the standard device and examine the directory. Bigfile2 is gone now, and bigfile is back. The reverse split operation copied bigfile from the BCV's fixed mirror to its moving mirror. The restore operation copied bigfile from the BCV's moving mirror to the standard.

```
mount /dev/dsk/c4t0d0s2 /stdmnt

ls -l /stdmnt
total 989984
-rw-r--r--   1 root     other    506601472 Mar 13 18:21 bigfile
drwx------   2 root     root          8192 Mar 13 17:41 lost+found
```

Example 2: Using the split -not_ready command

The following demonstration of the `split -not_ready` command uses a Windows 2000 server.

- The `sympd list` command displays physical devices available on the server devices. The example uses devices 59 (\PHYSICALDRIVE6) and 6A (\PHYSICALDRIVE11). The ellipsis (……) represents output that was truncated.

```
sympd list

Symmetrix ID: 000000003263

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

\\.\PHYSICALDRIVE1       0006 16B:1 02B:C0  Unprotected   N/Grp'd      RW     516
\\.\PHYSICALDRIVE2       0007 16B:1 01A:D3  Unprotected   N/Grp'd      RW     516
\\.\PHYSICALDRIVE3       0012 16B:1 02A:C1  BCV           N/Asst'd     RW     516
\\.\PHYSICALDRIVE4       002A 16B:1 02A:D2  RDF2          N/Grp'd      NR     516
\\.\PHYSICALDRIVE5       002B 16B:1 01B:C1  RDF2          N/Grp'd      NR     516
\\.\PHYSICALDRIVE6       0059 16B:1 02B:C1  Unprotected   N/Grp'd      RW     960
\\.\PHYSICALDRIVE7       005A 16B:1 01A:D2  Unprotected   N/Grp'd      RW     960
\\.\PHYSICALDRIVE8       005B 16B:1 01A:D1  Unprotected   Grp'd        RW     960
\\.\PHYSICALDRIVE9       0036 16B:1 02B:C3  RDF1-BCV      N/Asst'd     RW     516
\\.\PHYSICALDRIVE10      0037 16B:1 01A:D0  RDF1-BCV      N/Asst'd     RW     516
\\.\PHYSICALDRIVE11      006A 16B:1 01A:C0  2-Way BCV Mir N/Asst'd   RW     960
\\.\PHYSICALDRIVE12      006B 16B:1 02A:C0  2-Way BCV Mir N/Asst'd   RW     960
\\.\PHYSICALDRIVE13      006C 16B:1 01B:C0  2-Way BCV Mir Asst'd       RW     960
```

- The following commands create a device group and add one standard and one BCV to it.

```
symdg create ProdCgrp
symld -g ProdCgrp add dev 59
symbcv -g ProdCgrp associate dev 6A
```

- The first `symlabel list` command displays the signature of the standard; the second symlabel list command displays the BCV. Notice that the column for the defined device label reads N/A, while the actual label column contains a value. The Disk Administrator places that value on the physical disk when it first detects the device.

```
symlabel -g ProdCgrp -type wnt list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                      Device Label          Device
------------------------------------  ----------------------  --------------
Logical      Physical            Sym   Type  Defined  Actual    Sts  BCV <=> STD
------------------------------------  ----- -------- -------  --- ----------

DEV001     \\.\PHYSICALDRIVE6   0059  N/A   N/A      3F7AB9E8  RW  N/A
symlabel -g ProdCgrp -type wnt -bcv list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                      Device Label          Device
------------------------------------  -----------------------  -------------
Logical      Physical            Sym   Type  Defined  Actual    Sts  BCV <=> STD
------------------------------------  ----- -------- --------  --- ---------

BCV001       DRIVE11             006A  N/A   N/A      87654321  RW  NeverEstab
```

- The `symlabel define` command defines a label for the BCV but does not specify a label explicitly. Therefore, the existing *actual* label is read off the physical device and stored as the *defined* label in the SYMAPI database. The `symld list -offline` command allows the defined label to be read from the SYMAPI database without accessing the physical device.

```
symlabel -g ProdCgrp -bcv -type wnt define BCV001

symlabel -g ProdCgrp -bcv -type wnt -offline list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                      Device Label          Device
------------------------------------  ---------------------- -------------
Logical      Physical            Sym   Type  Defined  Actual    Sts  BCV <=> STD
------------------------------------  ---- -------- --------  --- ----------

BCV001       DRIVE11             006A  WNT  87654321 N/A       RW  NeverEstab
```

- The following commands initiate an establish operation followed by a `split -not_ready` operation. Wait for an establish operation to complete before attempting a split operation.

```
symmir -g ProdCgrp -full -exact -noprompt establish

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

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

```
symmir -g ProdCgrp -noprompt split -not_ready

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

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

- An examination of the BCV label reveals that its defined label is the original BCV label that was saved in the SYMAPI database. But its actual label matches that of the standard device (\PHYSICALDRIVE6). Although the BCV is split, the BCV status is NR (not ready).

```
symlabel -g ProdCgrp -bcv -type wnt list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                      Device Label          Device
------------------------------------  -----------------------  ----------------
Logical      Physical           Sym   Type  Defined  Actual    Sts BCV <=> STD
------------------------------------  ----- -------- --------  --- ---------

BCV001       DRIVE11            006A  WNT   87654321 3F7AB9E8  NR  Split
```

- The `symld relabel` command restores the original label to the BCV. The `-v` option provides more detail.

```
symld -g ProdCgrp relabel -bcv -v -noprompt

  Scanning device(s) for a 'RELABEL' operation (LOCAL).

    Device: 006A  (BCV     )           [SELECTED]

  STARTING a 'RELABEL' control operation.

    Relabel device(s) at the local Symmetrix..................Done.

  The 'RELABEL' control operation SUCCEEDED.

'Relabel' Device operation successfully completed for the device group.
```

- The following listing of the BCV label confirms that relabeling was successful.

```
symlabel -g ProdCgrp -bcv -type wnt list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                     Device Label           Device
------------------------------------  -----------------------  ------------
Logical       Physical           Sym   Type  Defined  Actual    Sts BCV <=> STD
------------------------------------  ----- -------- --------  --- ---------

BCV001        DRIVE11            006A  WNT   87654321 87654321  NR  Split
```

- The BCV becomes accessible to the host by issuing the `symld ready` command.

```
symld -g ProdCgrp -bcv -v -noprompt ready

  Scanning device(s) for a 'READY' operation (LOCAL).

    Device: 006A  (BCV     )           [SELECTED]

  STARTING a 'READY' control operation.

    Set device(s) Ready at local Symmetrix....................Done.

  The 'READY' control operation SUCCEEDED.

'Ready' Device operation successfully completed for the device group.
```

- An examination of the BCV reveals an RW (read/write) status, indicating that the ready action succeeded.

```
symlabel -g ProdCgrp -bcv -type wnt list

Device Group (DG) Name: ProdCgrp
DG's Type             : REGULAR
DG's Symmetrix ID     : 000000003263

            Device Name                     Device Label           Device
------------------------------------  ---------------------    ------------
Logical       Physical           Sym   Type  Defined  Actual    Sts BCV <=> STD
------------------------------------  ----- -------- --------  --- ---------

BCV001        DRIVE11             006A  WNT  87654321 87654321  RW  Split
```

Example 3: Using a composite group

This example uses Solutions Enabler Version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (sid 35 and 60). A composite group defined on the host contains a set of BCV pairs that spans the two Symmetrix arrays. You can control specific BCV pairs within the composite group as well as performing control operations on the entire group as shown here. For example, you can split one BCV pair without splitting the group's other pairs.)

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

  `symcg create TimeFinder -type regular`

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

  ```
  symcg -cg TimeFinder addall dev -range 56:6E -sid 35
  symcg -cg TimeFinder addall dev -range 14:27 -sid 60
  ```

- The following `symbcv` commands add to the composite group a range of BCV devices from each of the two source Symmetrix arrays.

  ```
  symbcv -cg TimeFinder associateall dev -range 182:19A -sid 35
  symbcv -cg TimeFinder associateall dev -range 3B6:3C9 -sid 60
  ```

- The `symmir establish` command creates optimized standard/BCV pairings among devices within each Symmetrix array and performs a full establish operation on them.

```
symmir -cg TimeFinder establish -full -optimize -noprompt

'Full Establish' operation execution is in progress for composite group
'TimeFinder'. Please wait...

'Full Establish' operation successfully initiated for composite group
'TimeFinder'.
```

- The `symmir query` command shows synchronization is in progress (SyncInProg) for the BCV pairs. Note that devices 56 and 59 are meta devices (as are their paired BCV devices), and only the meta head device is displayed.

```
symmir -cg TimeFinder query

Composite Group Name       :  TimeFinder
Composite Group Type       :  REGULAR
Number of Symmetrix Units  :  2
Number of RDF (RA) Groups  :  2

Symmetrix ID               : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                  State
-------------------------- --------------------------------- ------------
                    Inv.                                Inv.
Sym                 Tracks Sym              Att.       Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0056                     0 0182                *       344150 SyncInProg
0059                     0 0185                *       228145 SyncInProg
005B                     0 0187                *       120030 SyncInProg
005C                     0 0188                *       120181 SyncInProg
005D                     0 0189                *       120878 SyncInProg
005E                     0 018A                *       110078 SyncInProg
005F                     0 018B                *        41523 SyncInProg
0060                     0 018C                *        33900 SyncInProg
```

```
0061                      0 018D              *     123432 SyncInProg
0062                      0 018E              *      17008 SyncInProg
0063                      0 018F              *      53389 SyncInProg
0064                      0 0190              *      42266 SyncInProg
0065                      0 0191              *       7995 SyncInProg
0066                      0 0192              *      37293 SyncInProg
0067                      0 0193              *      45200 SyncInProg
0068                      0 0194              *      44386 SyncInProg
0069                      0 0195              *      45943 SyncInProg
006A                      0 0196              *      55380 SyncInProg
006B                      0 0197              *      55525 SyncInProg
006C                      0 0198              *      43544 SyncInProg
006D                      0 0199              *      48643 SyncInProg
006E                      0 019A              *      13424 SyncInProg

Symmetrix ID              : 000184500160    (Microcode Version: 5568)

      Standard Device                BCV Device                State
-------------------------- --------------------------------- ------------
                    Inv.                                Inv.
Sym                 Tracks Sym              Att.        Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0014                      0 03B6              *     131367 SyncInProg
0015                      0 03B7              *     120967 SyncInProg
0016                      0 03B8              *     114601 SyncInProg
0017                      0 03B9              *     101797 SyncInProg
0018                      0 03BA              *     122833 SyncInProg
0019                      0 03BB              *     117237 SyncInProg
001A                      0 03BC              *     115669 SyncInProg
001B                      0 03BD              *     121046 SyncInProg
001C                      0 03BF              *     132493 SyncInProg
001D                      0 03BE              *     125720 SyncInProg
001E                      0 03C0              *     113095 SyncInProg
001F                      0 03C1              *     103544 SyncInProg
0020                      0 03C2              *     124720 SyncInProg
0021                      0 03C3              *     109486 SyncInProg
0022                      0 03C4              *     109383 SyncInProg
0023                      0 03C5              *     122115 SyncInProg
0024                      0 03C6              *     133037 SyncInProg
0025                      0 03C7              *     110196 SyncInProg
0026                      0 03C8              *     118622 SyncInProg
0027                      0 03C9              *     118089 SyncInProg

Total               ------                             -------
  Track(s)                0                            4118330
  MB(s)                   0                             128698

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
```

- The `symmir split` command performs an instant split on all BCV pairs in the composite group.

```
symmir -cg TimeFinder split -instant -noprompt

'Split' operation execution is in progress for composite group 'TimeFinder'.
Please wait...

'Split' operation successfully executed for composite group 'TimeFinder'.
```

- The `symmir query` command with the `-bg` and `-percent` options shows the percentage completion of the split that is currently occurring in the background.

```
symmir -cg TimeFinder query -bg -percent

Composite Group Name      :  TimeFinder
Composite Group Type      :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID             : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                State     Split
-------------------------- ------------------------------ ----------- ----
                    Inv.                                 Inv.
Sym                 Tracks Sym                 Att.      Tracks STD <=> BCV   (%)
-------------------------- ------------------------------ ----------- ---
0056                     0 0182                 *             0 Split   (bg)  14
0059                     0 0185                 *             0 Split   (bg)  14
005B                     0 0187                 *             0 Split   (bg)  15
005C                     0 0188                 *             0 Split   (bg)  15
005D                     0 0189                 *             0 Split   (bg)  15
005E                     0 018A                 *             0 Split   (bg)  13
005F                     0 018B                 *             0 Split   (bg)  15
0060                     0 018C                 *             0 Split   (bg)  15
0061                     0 018D                 *             0 Split   (bg)  15
0062                     0 018E                 *             0 Split   (bg)  13
0063                     0 018F                 *             0 Split   (bg)  15
0064                     0 0190                 *             0 Split   (bg)  15
0065                     0 0191                 *             0 Split   (bg)  15
0066                     0 0192                 *             0 Split   (bg)  13
0067                     0 0193                 *             0 Split   (bg)  15
0068                     0 0194                 *             0 Split   (bg)  15
0069                     0 0195                 *             0 Split   (bg)  15
006A                     0 0196                 *             0 Split   (bg)  13
006B                     0 0197                 *             0 Split   (bg)  15
006C                     0 0198                 *             0 Split   (bg)  15
006D                     0 0199                 *             0 Split   (bg)  15
006E                     0 019A                 *             0 Split   (bg)  13
Symmetrix ID              : 000184500160    (Microcode Version: 5568)

      Standard Device                BCV Device                State     Split
-------------------------- ------------------------------- ---------- ---
                    Inv.                                 Inv.
Sym                 Tracks Sym                 Att.      Tracks STD <=> BCV   (%)
-------------------------- ------------------------------- ----------- ---
0014                     0 03B6                 *             0 Split   (bg)   5
0015                     0 03B7                 *             0 Split   (bg)   5
0016                     0 03B8                 *             0 Split   (bg)   5
0017                     0 03B9                 *             0 Split   (bg)   2
0018                     0 03BA                 *             0 Split   (bg)   2
0019                     0 03BB                 *             0 Split   (bg)   2
001A                     0 03BC                 *             0 Split   (bg)   2
001B                     0 03BD                 *             0 Split   (bg)   2
001C                     0 03BF                 *             0 Split   (bg)   5
001D                     0 03BE                 *             0 Split   (bg)   2
001E                     0 03C0                 *             0 Split   (bg)   9
```

```
001F                    0 03C1                    *          0 Split   (bg)    8
0020                    0 03C2                    *          0 Split   (bg)    6
0021                    0 03C3                    *          0 Split   (bg)    6
0022                    0 03C4                    *          0 Split   (bg)   18
0023                    0 03C5                    *          0 Split   (bg)   17
0024                    0 03C6                    *          0 Split   (bg)    9
0025                    0 03C7                    *          0 Split   (bg)    8
0026                    0 03C8                    *          0 Split   (bg)    6
0027                    0 03C9                    *          0 Split   (bg)    7

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

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
(bg): The paired BCV device is splitting in the background.
```

- The `symmir verify` command checks the background split at 60-second intervals until the split operation on the composite group has completed.

`symmir -cg TimeFinder verify -bg -split -i 60`

```
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
All devices in Composite Group 'TimeFinder' have finished splitting in the
background.
```

11

Performing Consistent Splits with TimeFinder Family

This chapter provides examples on splitting off a consistent, DBMS-restartable BCV copy of your database without having to quiesce or shut down the database first.

The following examples illustrate TimeFinder functionality in versions of EMC Solutions Enabler up to Version 6.0 running on Symmetrix arrays using Enginuity Versions up to 5x71.

Example 1: Creating an RDBMS-restartable copy on a local Symmetrix array

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

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

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

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

sympd list -powerpath

```
Symmetrix ID: 000185500814

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

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

                         0286       02B:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d2            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d2            - 04A:0      - -              -            -        -

                         0287       01B:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d3            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d3            - 04A:0      - -              -            -        -

                         0288       16A:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d4            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d4            - 04A:0      - -              -            -        -

                         0289       15A:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d5            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d5            - 04A:0      - -              -            -        -

                         028A       02A:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d6            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d6            - 04A:0      - -              -            -        -

                         028B       01A:C5 2-Way Mir      N/Grp'd      RW    4315
/dev/rdsk/c0t1d7            - 13A:0      - -              -            -        -
/dev/rdsk/c1t1d7            - 04A:0      - -              -            -        -
```

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

```
sympd list

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

/dev/rdsk/c0t0d0        039A 13B:0 16B:D5  BCV           N/Asst'd     RW    4315
/dev/rdsk/c0t0d1        039B 13B:0 01A:D5  BCV           N/Asst'd     RW    4315
/dev/rdsk/c0t0d2        039C 13B:0 16B:C5  BCV           N/Asst'd     RW    4315
/dev/rdsk/c0t0d3        039D 13B:0 15B:C5  BCV           N/Asst'd     RW    4315
/dev/rdsk/c0t0d4        039E 13B:0 02B:C5  BCV           N/Asst'd     RW    4315
/dev/rdsk/c0t0d5        039F 13B:0 01B:C5  BCV           N/Asst'd     RW    4315
```

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

```
symdg create oracle
symld -g oracle addall dev -range 286:28B
symbcv -g oracle associateall dev -range 39A:39F
```

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

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

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

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

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

```
symmir -g oracle verify -i 30

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

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

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

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

  `export SYMCLI_RDB_CONNECT=system/manager`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
symmir -g oracle establish -noprompt

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

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

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

```
symmir -g oracle verify -i 30

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
symmir -g oracle establish -noprompt

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

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

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

```
symmir -g oracle verify -i 30

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

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

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

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

```
symmir -g oracle split -instant -noprompt -ppath /dev/rdsk/c0t1d2
/dev/rdsk/c0t1d3 /dev/rdsk/c0t1d4 /dev/rdsk/c0t1d5 /dev/rdsk/c0t1d6
/dev/rdsk/c0t1d7

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

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

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

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

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

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

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

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

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

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

```
symmir -g oracle establish -noprompt

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

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

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

```
symmir -g oracle verify -i 30

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

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

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

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

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

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

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

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

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

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

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

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

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

```
symmir -g oracle establish -noprompt

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

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

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

```
symmir -g oracle verify -i 30

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Example 2: Creating an RDBMS-restartable BCV copy on a remote Symmetrix array using PowerPath

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

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

To perform a consistent split in an SRDF-based environment, you must run SRDF/S with Synchronous replication.

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

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

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

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

      000185500814 Local       8530   5568       4088       450      1052
      000185500505 Remote      8530   5568       8184         0       950
  ```

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

sympd list -powerpath
```
Symmetrix ID: 000185500814
                          P O W E R P A T H      D E V I C E S
        Device Name              Directors                   Device
---------------------------- ------------ -------------------------------------
                                                                          Cap
Physical                 Sym  SA :P DA :IT Config         Attribute    Sts   (MB)
---------------------------- ------------ -------------------------------------

                         03C2       16B:D5 RDF1           N/Grp'd      RW    4315
/dev/rdsk/c4t0d102s2        - 05B:0     - -              -            -        -
/dev/rdsk/c5t0d102s2        - 12B:0     - -              -            -        -

                         03C3       15B:D5 RDF1           N/Grp'd      RW    4315
/dev/rdsk/c4t0d103s2        - 05B:0     - -              -            -        -
/dev/rdsk/c5t0d103s2        - 12B:0     - -              -            -        -
                         03C4       02B:D5 RDF1           N/Grp'd      RW    4315
/dev/rdsk/c4t0d104s2        - 05B:0     - -              -            -        -
/dev/rdsk/c5t0d104s2        - 12B:0     - -              -            -        -

                         03C5       01B:D5 RDF1           N/Grp'd      RW    4315
/dev/rdsk/c4t0d105s2        - 05B:0     - -              -            -        -
/dev/rdsk/c5t0d105s2        - 12B:0     - -              -            -        -

                         03C6       16A:D5 RDF1           N/Grp'd      RW    4315
/dev/rdsk/c4t0d106s2        - 05B:0     - -              -            -        -
/dev/rdsk/c5t0d106s2        - 12B:0     - -              -            -        -
.................................................................................
```

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

`symdev list -bcv -sid 505`

```
Symmetrix ID: 000185500505

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

038A Not Visible             ???:? 02B:C5 BCV           N/Asst'd     RW    4315
038B Not Visible             ???:? 01B:C5 BCV           N/Asst'd     RW    4315
038C Not Visible             ???:? 16A:C5 BCV           N/Asst'd     RW    4315
038D Not Visible             ???:? 15A:C5 BCV           N/Asst'd     RW    4315
038E Not Visible             ???:? 02A:D4 BCV           N/Asst'd     RW    4315
```

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

```
symdg create oracle -type rdf1
symld -g oracle addall dev -range 3C2:3C6
symbcv -g oracle associateall dev -range 38A:38E -rdf
```

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

```
symrdf -g oracle establish -full -noprompt

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

    Suspend RDF link(s).......................................Done.
    Mark target (R2) devices for full copy from source (R1)...Started.
    Device: 03C2 ............................................ Marked.
    Device: 03C3 ............................................ Marked.
    Device: 03C4 ............................................ Marked.
    Device: 03C5 ............................................ Marked.
    Device: 03C6 ............................................ Marked.
    Mark target (R2) devices for full copy from source (R1)...Done.
    Merge device track tables between source and target.......Started.
    Device: 03C2 ............................................ Merged.
    Device: 03C3 ............................................ Merged.
    Device: 03C4 ............................................ Merged.
    Device: 03C5 ............................................ Merged.
    Device: 03C6 ............................................ Merged.
    Merge device track tables between source and target.......Done.
    Read/Write Enable device(s) on SA at source (R1)..........Done.
    Resume RDF link(s)........................................Done.

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

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

```
symrdf -g oracle verify -i 30

NONE of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
symmir -g oracle verify -split -bg -i 10 -rdf

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

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

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

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

```
symmir -g oracle establish -rdf -noprompt

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

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

- The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

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

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

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

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

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

```
symmir -g oracle split -instant -rdf -noprompt -ppath
/dev/rdsk/c4t0d102s2 /dev/rdsk/c4t0d103s2 /dev/rdsk/c4t0d104s2
/dev/rdsk/c4t0d105s2 /dev/rdsk/c4t0d106s2

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

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

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

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

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

```
symmir -g oracle verify -split -bg -i 10 -rdf

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

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

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

- The `symmir establish` command re-establishes incrementally the five remote BCV pairs in the device group that are in the Split state.

```
symmir -g oracle establish -rdf -noprompt

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

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

- The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

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

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

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

Example 3: ECA consistent split of a composite group

This example was performed using Solutions Enabler Version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (sid 35 and 60). A composite group defined on the host contains a set of BCV pairs that spans the two Symmetrix arrays. The devices include standard devices and BCV devices. However, this same functionality can be applied to TimeFinder snap pairs and clone pairs.

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

  ```
  symcg create TimeFinder -type regular
  ```

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

  ```
  symcg -cg TimeFinder addall dev -range 56:6E -sid 35
  symcg -cg TimeFinder addall dev -range 14:27 -sid 60
  ```

- The following `symbcv` commands add to the composite group a range of BCV devices from each of the two source Symmetrix arrays.

  ```
  symbcv -cg TimeFinder associateall dev -range 182:19A -sid 35
  symbcv -cg TimeFinder associateall dev -range 3B6:3C9 -sid 60
  ```

- The `symmir establish` command creates optimized standard/BCV pairings among devices within each Symmetrix array and performs a full establish operation on them.

  ```
  symmir -cg TimeFinder establish -full -optimize -noprompt

  'Full Establish' operation execution is in progress for composite
  group
  'TimeFinder'. Please wait...

  'Full Establish' operation successfully initiated for composite
  group
  'TimeFinder'.
  ```

- The `symmir query` command shows synchronization is in progress (SyncInProg) for the BCV pairs. Devices 0056 and 0059 are meta devices (as are their paired BCV devices), and only the meta head device is displayed.

symmir -cg TimeFinder query

```
Composite Group Name       :  TimeFinder
Composite Group Type       :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID              : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                  State
-------------------------- --------------------------------- ------------
                    Inv.                                 Inv.
Sym               Tracks  Sym                 Att.     Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0056                     0 0182                  *      344150 SyncInProg
0059                     0 0185                  *      228145 SyncInProg
005B                     0 0187                  *      120030 SyncInProg
005C                     0 0188                  *      120181 SyncInProg
005D                     0 0189                  *      120878 SyncInProg
005E                     0 018A                  *      110078 SyncInProg
005F                     0 018B                  *       41523 SyncInProg
0060                     0 018C                  *       33900 SyncInProg
0061                     0 018D                  *      123432 SyncInProg
0062                     0 018E                  *       17008 SyncInProg
0063                     0 018F                  *       53389 SyncInProg
0064                     0 0190                  *       42266 SyncInProg
0065                     0 0191                  *        7995 SyncInProg
0066                     0 0192                  *       37293 SyncInProg
0067                     0 0193                  *       45200 SyncInProg
0068                     0 0194                  *       44386 SyncInProg
0069                     0 0195                  *       45943 SyncInProg
006A                     0 0196                  *       55380 SyncInProg
006B                     0 0197                  *       55525 SyncInProg
006C                     0 0198                  *       43544 SyncInProg
006D                     0 0199                  *       48643 SyncInProg
006E                     0 019A                  *       13424 SyncInProg

Symmetrix ID              : 000184500160    (Microcode Version: 5568)

      Standard Device                BCV Device                  State
-------------------------- --------------------------------- ------------
                    Inv.                                 Inv.
Sym               Tracks  Sym                 Att.     Tracks STD <=> BCV
-------------------------- --------------------------------- ------------
0014                     0 03B6                  *      131367 SyncInProg
0015                     0 03B7                  *      120967 SyncInProg
0016                     0 03B8                  *      114601 SyncInProg
0017                     0 03B9                  *      101797 SyncInProg
0018                     0 03BA                  *      122833 SyncInProg
0019                     0 03BB                  *      117237 SyncInProg
001A                     0 03BC                  *      115669 SyncInProg
001B                     0 03BD                  *      121046 SyncInProg
001C                     0 03BF                  *      132493 SyncInProg
001D                     0 03BE                  *      125720 SyncInProg
001E                     0 03C0                  *      113095 SyncInProg
001F                     0 03C1                  *      103544 SyncInProg
0020                     0 03C2                  *      124720 SyncInProg
0021                     0 03C3                  *      109486 SyncInProg
0022                     0 03C4                  *      109383 SyncInProg
0023                     0 03C5                  *      122115 SyncInProg
0024                     0 03C6                  *      133037 SyncInProg
0025                     0 03C7                  *      110196 SyncInProg
```

```
0026                       0 03C8                     *      118622 SyncInProg
0027                       0 03C9                     *      118089 SyncInProg

Total               ------                                  -------
  Track(s)                 0                                4118330
  MB(s)                    0                                 128698

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
```

- The `symmir split` command performs an ECA consistent split on all BCV pairs in the composite group.

  `symmir -cg TimeFinder split -consistent -noprompt`

  ```
  'Split' operation execution is in progress for composite group
  'TimeFinder'. Please wait...

  'Split' operation successfully executed for composite group
  'TimeFinder'.
  ```

- The `symmir query` command with the `-bg` and `-percent` options shows the percentage completion of the split that is currently occurring in the background.

`symmir -cg TimeFinder query -bg -percent`

```
Composite Group Name      :  TimeFinder
Composite Group Type      :  REGULAR
Number of Symmetrix Units :  2
Number of RDF (RA) Groups :  2

Symmetrix ID              : 000187900035    (Microcode Version: 5670)

      Standard Device                BCV Device                 State      Split
-------------------------- --------------------------------- ------------ -----
                    Inv.                               Inv.
Sym                 Tracks Sym              Att.       Tracks STD <=> BCV   (%)
-------------------------- --------------------------------- ------------ -----
0056                     0 0182               *             0 Split   (bg)  14
0059                     0 0185               *             0 Split   (bg)  14
005B                     0 0187               *             0 Split   (bg)  15
005C                     0 0188               *             0 Split   (bg)  15
005D                     0 0189               *             0 Split   (bg)  15
005E                     0 018A               *             0 Split   (bg)  13
005F                     0 018B               *             0 Split   (bg)  15
0060                     0 018C               *             0 Split   (bg)  15
0061                     0 018D               *             0 Split   (bg)  15
0062                     0 018E               *             0 Split   (bg)  13
0063                     0 018F               *             0 Split   (bg)  15
0064                     0 0190               *             0 Split   (bg)  15
0065                     0 0191               *             0 Split   (bg)  15
0066                     0 0192               *             0 Split   (bg)  13
0067                     0 0193               *             0 Split   (bg)  15
0068                     0 0194               *             0 Split   (bg)  15
0069                     0 0195               *             0 Split   (bg)  15
006A                     0 0196               *             0 Split   (bg)  13
006B                     0 0197               *             0 Split   (bg)  15
006C                     0 0198               *             0 Split   (bg)  15
006D                     0 0199               *             0 Split   (bg)  15
006E                     0 019A               *             0 Split   (bg)  13

Symmetrix ID              : 000184500160    (Microcode Version: 5568)
```

```
      Standard Device                BCV Device                 State      Split
-------------------------- --------------------------------- ------------ -----
                    Inv.                                Inv.
Sym                 Tracks Sym                 Att.      Tracks STD <=> BCV   (%)
-------------------------- --------------------------------- ------------ -----
0014                     0 03B6                  *           0 Split   (bg)    5
0015                     0 03B7                  *           0 Split   (bg)    5
0016                     0 03B8                  *           0 Split   (bg)    5
0017                     0 03B9                  *           0 Split   (bg)    2
0018                     0 03BA                  *           0 Split   (bg)    2
0019                     0 03BB                  *           0 Split   (bg)    2
001A                     0 03BC                  *           0 Split   (bg)    2
001B                     0 03BD                  *           0 Split   (bg)    2
001C                     0 03BF                  *           0 Split   (bg)    5
001D                     0 03BE                  *           0 Split   (bg)    2
001E                     0 03C0                  *           0 Split   (bg)    9
001F                     0 03C1                  *           0 Split   (bg)    8
0020                     0 03C2                  *           0 Split   (bg)    6
0021                     0 03C3                  *           0 Split   (bg)    6
0022                     0 03C4                  *           0 Split   (bg)   18
0023                     0 03C5                  *           0 Split   (bg)   17
0024                     0 03C6                  *           0 Split   (bg)    9
0025                     0 03C7                  *           0 Split   (bg)    8
0026                     0 03C8                  *           0 Split   (bg)    6
0027                     0 03C9                  *           0 Split   (bg)    7

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

Legend:

(*): The paired BCV device is associated with this composite group.
(p): The paired BCV device was restored using the protect option.
(a): All mirrors of this BCV were moved to the STD.
(bg): The paired BCV device is splitting in the background.
```

- The `symmir verify` command checks the background split at 60-second intervals until the split operation on the composite group has completed.

symmir -cg TimeFinder verify -bg -split -i 60

```
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
Not all devices in Composite Group 'TimeFinder' have finished splitting in the
background.
All devices in Composite Group 'TimeFinder' have finished splitting in the
background.
```

Example 4: Performing splits with database assist

Performing splits without PowerPath or ECA is similar for most supported databases in suspending writes to a database momentarily while an instant split occurs (refer to "Example 4: Performing splits with database assist" on page 416). The following examples show how to perform splits this way for an SQL 2000 database and an Oracle database.

Splitting BCV pairs using SQL 2000 database assist

This setup consists of a Windows 2000 host connected to a Symmetrix array (sid 505). SQL Server 2000, Enterprise Edition, software is installed on the host. An SQL 2000 database called SQL2000a was created on Symmetrix devices. A device group called sqlserv was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

The `symmir query` command displays the configuration and status of the BCV pairs that hold the database. All BCV pairs are currently in the Synchronized state.

```
symmir -g sqlserv query

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

     Standard Device                  BCV Device                  State
-------------------------- ------------------------------------- ------------
                    Inv.                                   Inv.
Logical         Sym  Tracks Logical                 Sym    Tracks STD <=> BCV
-------------------------- ------------------------------------- ------------

DEV001          0000      0 BCV001                  00C8 *      0 Synchronized
DEV002          0001      0 BCV002                  00C9 *      0 Synchronized
DEV003          0002      0 BCV003                  00CA *      0 Synchronized
DEV004          0003      0 BCV004                  00CB *      0 Synchronized
DEV005          0004      0 BCV005                  00CC *      0 Synchronized
DEV006          0005      0 BCV006                  00CD *      0 Synchronized
DEV007          0006      0 BCV007                  00CE *      0 Synchronized
DEV008          0007      0 BCV008                  00CF *      0 Synchronized
DEV009          0008      0 BCV009                  00D0 *      0 Synchronized
DEV010          0009      0 BCV010                  00D1 *      0 Synchronized
DEV011          000A      0 BCV011                  00D2 *      0 Synchronized
DEV012          000B      0 BCV012                  00D3 *      0 Synchronized
DEV013          000C      0 BCV013                  00D4 *      0 Synchronized
DEV014          000D      0 BCV014                  00D5 *      0 Synchronized
DEV015          000E      0 BCV015                  00D6 *      0 Synchronized
DEV016          000F      0 BCV016                  00D7 *      0 Synchronized
DEV017          0010      0 BCV017                  00D8 *      0 Synchronized
DEV018          0011      0 BCV018                  00D9 *      0 Synchronized
DEV019          0012      0 BCV019                  00DA *      0 Synchronized
DEV020          0013      0 BCV020                  00DB *      0 Synchronized
DEV021          0014      0 BCV021                  00DC *      0 Synchronized

Total                 -------                                 -------
  Track(s)                  0                                       0
  MB(s)                   0.0                                     0.0
Legend:

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

- For SYMCLI to access a specified database, set the SYMCLI_RDB_CONNECT environment variable to the username and password of the system administrator's account. The export command sets this variable to a username of `system` and a password of `manager`, allowing a local connection.

  ```
  set SYMCLI_RDB_CONNECT=system/manager
  ```

- The `symioctl begin snapshot` command sends the SQL command BACKUP DATABASE TO VIRTUAL_DEVICE WITH SNAPSHOP to SQL Server, which begins the snapshot backup and suspends writes to the database named SQL2000a. It also creates a savefile named sql.save.

  ```
  symioctl begin snapshot SQL2000a savefile sql.save -type sqlserver
  -nop
  ```

- The `symmir split` command performs an instant split on all BCV pairs in the device group sqlserv.

```
symmir -g sqlserv split -instant -noprompt

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

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

- The `symioctl end snapshot` command causes the SQL command BACKUP DATABASE to complete. Writes to the database can resume, and the snapshot metadata is saved to the save file.

  ```
  symioctl end snapshot -type sqlserver SQL2000a -noprompt
  ```

Splitting BCV pairs using Oracle database assist

The hardware setup for this example consists of a Solaris host (api179) connected to a local Symmetrix array. Oracle 8.1.7.0.0 software is installed on api179. A device group called oracle was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

For SYMCLI to access a specified database, set the SYMCLI_RDB_CONNECT environment variable to the username and password of the system administrator's account. The first `export` command sets this variable to a username of `system` and a password of `manager`, allowing a local connection to Oracle. The `export ORACLE_HOME` command specifies the location of the Oracle binaries. The `export ORACLE_SID` command specifies the database instance name. (Other RDBMS database systems use environment variables specific to their system. For example, Sybase uses the variable SYBASE for the location of the Sybase binaries and uses DSQUERY to specify the server name.)

```
export SYMCLI_RDB_CONNECT=system/manager
export ORACLE_HOME=/disks/symapidvt/oraclehome/api179
export ORACLE_SID=api179
```

- The `symrdb list` command allows you to test basic database connectivity.

  ```
  symrdb list -type oracle

  DATABASE NAMES (ORACLE 8.1.7.0.0):

    Database Name
    -------------

    api179
  ```

- The `symioctl begin backup` command places all tables in the Oracle database (api179) in hot backup mode.

  `symioctl -type oracle begin backup -noprompt`

- The `symioctl freeze` command suspends writes to the Oracle database (api179).

  `symioctl -type oracle freeze -noprompt`

- The `symmir split` command performs an instant split on all BCV pairs in the device group called oracle.

  `symmir -g oracle split -instant -noprompt`

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

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

- The `symioctl thaw` command allows writes to the database to resume.

  `symioctl -type oracle thaw -noprompt`

- The `symioctl end backup` command terminates hot backup mode.

  `symioctl -type oracle end backup -noprompt`

A

SRDF State Rules Reference

This appendix describes the applicable SRDF pair states that rule the TimeFinder/Clone and TimeFinder/Snap copy session operations.

SRDF pair states

Certain TimeFinder/Clone and TimeFinder/Snap copy operations are not allowed within Symmetrix ~~storage~~ arrays employing the Symmetrix Remote Data Facility (SRDF) for remote mirroring as these operations can conflict with one another. The availability of some actions depends on the current state of SRDF pairs. This chapter describes each of the SRDF pair states and what Clone and Snap operations that are available within that state.

Table 27 provides a description the various SRDF pair states.

Table 27 SRDF pair states

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

State rules for TimeFinder/Clone operations

This section identifies the `symclone` copy actions that are available for use within each of the SRDF pair states.

Consistent

Table 28 identifies the `symclone` copy operations available for use when SRDF pairs are in the Consistent pair state for asynchronous mode.

Table 28 **TimeFinder/Clone operations for Consistent pair states**

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔[a, b, c]	✔[d]	
Activate	✔[a]	✔[a, b, c]	✔	
Establish	✔[a]	✔[a, b, c]		
Terminate	✔	✔	✔	✔
Restore	✔[a]	✔[a, b, c]		
Split	✔	✔[a, b, c]		

a. Action is not allowed if there are local R1 invalids or remote R2 invalids.

b. The force option (`-force`) must be applied.

c. Action is not allowed with CopyOnAccess.

d. The precopy option (`-precopy`) must be applied.

Failed over

Table 29 identifies the `symclone` copy operations available for use when SRDF pairs are in the Failed Over pair state.

Table 29 **TimeFinder/Clone operations for Failed Over pair states**

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔		✔	✔[a, b]
Activate/Establish	✔		✔	✔[a, b]
Terminate	✔	✔	✔	✔
Restore	✔		✔	✔[a, b]
Split	✔		✔	✔[a, b]

a. Action not allowed if the R2 target is in SRDF/Asynchronous mode.

b. Action is not allowed if the target device is an R2 larger than the R1.

Invalid

Table 30 identifies the `symclone` copy operations available for use when SRDF pairs are in the Invalid pair state.

Table 30 TimeFinder/Clone operations for Invalid pair states

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate				
Activate/Establish				
Terminate	✔	✔	✔	✔
Restore				
Split				

Partitioned

Table 31 identifies the `symclone` copy operations available for use when SRDF pairs are in the Partitioned pair state.

Table 31 TimeFinder/Clone operations for partitioned pair states

TimeFinder/Clone Action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔	✔	✔[a, b]
Activate/Establish	✔	✔	✔	✔[a, b]
Terminate	✔	✔	✔	✔
Restore	✔	✔	✔	✔[a, b]
Split	✔	✔	✔	✔[a, b]

a. Action is not allowed if the R2 target is in SRDF/Asynchronous mode.

b. Action is not allowed if the target device is an R2 larger than the R1.

R1 updated

Table 32 identifies the `symclone` copy operations available for use when SRDF pairs are in the R1 Updated pair state.

Table 32 TimeFinder/Clone operations for R1 Updated pair states

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔		✔	
Activate/Establish	✔[a]		✔	
Terminate	✔	✔	✔	✔
Restore	✔[a]		✔	
Split	✔		✔	

a. Action is not allowed if the `-consistent` option was specified.

R1 update in progress

Table 33 identifies the `symclone` copy operations available for use when SRDF pairs are in the R1 UpdInProg pair state.

Table 33 **TimeFinder/Clone operations for R1 Updinprog pair states**

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate			✔	
Activate/Establish	✔[a]		✔	
Terminate	✔	✔	✔	✔
Restore	✔		✔	
Split			✔	

a. Action is not allowed if the `-consistent` option was specified.

Split

Table 34 identifies the `symclone` copy operations available for use when SRDF pairs are in the Split pair state.

Table 34 **TimeFinder/Clone operations for Split pair states**

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔	✔	✔[a, b]
Activate/Establish	✔	✔	✔	✔[a, b]
Terminate	✔	✔	✔	✔
Restore	✔	✔	✔	✔[a, b]
Split	✔	✔	✔	✔[a, b]

a. Action is not allowed if the R2 target is in SRDF/Asynchronous mode.

b. Action is not allowed if the target device is an R2 larger than the R1.

Suspended

Table 35 identifies the `symclone` copy operations available for use when SRDF pairs are in the Suspended pair state.

Table 35 TimeFinder/Clone operations for suspended pair states

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔	✔	✔[a, b]
Activate/Establish	✔	✔	✔	✔[a, b]
Terminate	✔	✔	✔	✔
Restore	✔	✔	✔	✔[a, b]
Split	✔	✔	✔	✔[a, b]

a. Action is not allowed if the R2 target is in SRDF/Asynchronous mode.

b. Action is not allowed if the target device is an R2 larger than the R1.

Synchronized

Table 36 identifies the `symclone` copy operations available for use when SRDF pairs are in the Synchronized pair state.

Table 36 TimeFinder/Clone operations for synchronized pair states

TimeFinder/Clone Action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔[a, b]	✔	
Activate/Establish	✔	✔[a, b]	✔	
Terminate	✔	✔	✔	✔
Restore	✔	✔[a, b]	✔	
Split	✔	✔[a, b]	✔	

a. The force option (`-force`) must be applied.

b. Action is not allowed with CopyOnAccess.

Sync in progress

Table 37 identifies the `symclone` copy operations available for use when SRDF pairs are in the SyncInProg pair state.

Table 37 TimeFinder/Clone operations for SyncInProg pair states

TimeFinder/Clone action	R1 copy source	R1 copy target	R2 copy source	R2 copy target
Create/Recreate	✔	✔[a, c, d]	✔	
Activate/Establish	✔[a]	✔[a, c, d]	✔[b]	
Terminate	✔	✔	✔	✔
Restore	✔[a]	✔[a, c, d]	✔[b]	
Split	✔	✔[a, c, d]	✔	

a. Action is not allowed if there are local R1 invalids or remote R2 invalids.

b. Action is not allowed if there are remote R1 invalids or local R2 invalids.

c. The force option (`-force`) must be applied.

d. Action is not allowed with CopyOnAccess.

State rules for TimeFinder/Snap operations

This section identifies what symsnap copy actions are available for use within each of the SRDF pair states.

Consistent

Table 38 identifies the symsnap copy operations available for use when SRDF pairs are in the Consistent pair state.

Table 38 **TimeFinder/Snap operations for consistent pair states**

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A		N/A
Activate	✔[a]	N/A		N/A
Terminate	✔	N/A	✔[c]	N/A
Incremental restore to source	✔[a, b]	N/A		N/A
Incremental restore to a split BCV or full restore to any device	✔	✔[a, b]		

a. Action is not allowed if there are local R1 invalids or remote R2 invalids.

b. The force option (-force) must be applied.

c. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Failed Over

Table 39 identifies the symsnap copy operations available for use when SRDF pairs are in the Failed Over pair state.

Table 39 **TimeFinder/Snap operations for failed over pair states**

TimeFinder/Snap Action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source	✔	N/A	✔[b]	N/A
Incremental restore to a split BCV or full restore to any device	✔		✔[b]	✔[b, a]

a. Action is not allowed if there are remote R1 invalids or local R2 invalids.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Invalid

Table 40 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Invalid pair state.

Table 40 **TimeFinder/Snap operations for Invalid pair states**

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create		N/A		N/A
Activate		N/A		N/A
Terminate	✔	N/A	✔[a]	N/A
Incremental restore to source		N/A		N/A
Incremental restore to a split BCV or full restore to any device				

a. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Partitioned

Table 41 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Partitioned pair state.

Table 41 **TimeFinder/Snap operations for partitioned pair states**

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source	✔	N/A	✔[b]	N/A
Incremental restore to a split BCV or full restore to any device	✔	✔	✔[b]	✔[b, a]

a. Action is not allowed if there are remote R1 invalids or local R2 invalids.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

R1 updated

Table 42 identifies the `symsnap` copy operations available for use when SRDF pairs are in the R1 Updated pair state.

Table 42 TimeFinder/Snap operations for R1 updated pair states

TimeFinder/Snap Action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔[a]	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source		N/A	✔[b]	
Incremental restore to a split BCV or full restore to any device	✔[a]		✔[b]	

a. Action is not allowed if there are local R1 invalids or remote R2 invalids.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

R1 update in progress

Table 43 identifies the `symsnap` copy operations available for use when SRDF pairs are in the R1 UpdInProg pair state.

Table 43 TimeFinder/Snap operations for R1 Updinprog pair states

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔[a]	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source		N/A	✔[b]	N/A
Incremental restore to a split BCV or full restore to any device	✔[a]		✔[b]	

a. Action is not allowed if there are local R1 invalids or remote R2 invalids.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Split

Table 34 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Split pair state.

Table 44 TimeFinder/Snap operations for Split pair states

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source	✔	N/A	✔[b]	N/A
Incremental restore to a split BCV or full restore to any device	✔	✔	✔[b]	✔[b, a]

a. Action is not allowed if there are remote R1 invalids or local R2 invalids.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Suspended

Table 45 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Suspended pair state.

Table 45 TimeFinder/Snap operations for Suspended pair states

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[a]	N/A
Activate	✔	N/A	✔[a]	N/A
Terminate	✔	N/A	✔[a]	N/A
Incremental restore to source	✔	N/A	✔[a]	N/A
Incremental restore to a split BCV or full restore to any device	✔	✔	✔[a]	✔[a, b]

a. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

b. Action is not allowed if the target device is an R2 larger than the R1.

Synchronized

Table 46 identifies the symsnap copy operations available for use when SRDF pairs are in the Synchronized pair state.

Table 46 TimeFinder/Snap operations for Synchronized pair states

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[b]	N/A
Activate	✔	N/A	✔[b]	N/A
Terminate	✔	N/A	✔[b]	N/A
Incremental restore to source	✔[a]	N/A		N/A
Incremental restore to a split BCV or full restore to any device	✔	✔[a]	✔[b]	

a. The force option (-force) must be applied.

b. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

Sync in progress

Table 47 identifies the symsnap copy operations available for use when SRDF pairs are in the SyncInProg pair state.

Table 47 TimeFinder/Snap operations for Syncinprog pair states

TimeFinder/Snap action	R1 snap source	R1 snap target	R2 snap source	R2 snap target
Create	✔	N/A	✔[d]	N/A
Activate	✔[a]	N/A	✔[b, d]	N/A
Terminate	✔	N/A	✔[d]	N/A
Incremental restore to source	✔[a, c]	N/A		N/A
Incremental restore to a split BCV or full restore to any device	✔	✔[a, c]	✔[d]	

a. Action is not allowed if there are local invalids on the R1 side or remote invalids owed to the R1 on the R2 side.

b. Action is not allowed if there are remote invalids owed to the R2 on the R1 side or local invalids on the R2 side.

c. The force option (`-force`) must be applied.

d. Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.

B

TimeFinder State Reference

This appendix describes the applicable TimeFinder pair states that rule TimeFinder/Clone and TimeFinder/Snap copy session operations.

TimeFinder pair states

Certain TimeFinder/Clone and TimeFinder/Snap copy operations are not allowed within Symmetrix arrays employing TimeFinder mirroring. The availability of some actions depends on the current state of the TimeFinder BCV pairs. This appendix describes each of the TimeFinder pair states and which TimeFinder/Clone and TimeFinder/Snap operations that are available within that state.

Table 48 provides a description the various TimeFinder pair states.

Table 48 TimeFinder pair states

BCV pair state	Description
Sync In Progress	When the Establish action is executed, data is copied from the standard device to the BCV device until both devices contain identical data.
Synchronized	The BCV and standard devices have identical data. Any changes to the standard device are also written to the BCV. The BCV is unavailable to the host for BC processing.
Split in Progress	The BCV devices are in the process of being separated or split from the standard devices.
Split	The BCV devices are completely separated or split from the standard devices, allowing each device to be accessed separately by the host. No restrictions.
Split No Incremental	The BCV devices are completely separated or split from the standard devices, but cannot be incrementally established or restored. No Restrictions.
Restore In Progress	When the restore action is executed, data is copied from the BCV to the standard device until both devices contain identical data.
Restored	The BCV and standard devices have identical data, although the data was originally on the BCV before being synchronized. Any changes to the standard device are also written to the BCV. The BCV is unavailable to the host for BC processing.
Split Before Sync	The split occurred when a BCV device was synchronizing. The BCV device is separated from the standard device although the BCV device is not completely synchronized. No Restrictions.
Split Before Restore	The split occurred when a BCV device was being restored to a standard device. The BCV device is separated from the standard device although the standard device is not completely synchronized. No Restrictions.

Pair states ruling TimeFinder/Clone operations

Do not perform any TimeFinder/Clone action while the device(s) are in any of the following states:

- Split In Progress
- Restore In Progress

You can perform any TimeFinder/Clone action while the device(s) are in any of the following states:

- Split
- Split No Incremental
- Split Before Sync
- Split Before Restore

Create action

Table 49 identifies which `symclone` copy **create** actions are available for use for each of the TimeFinder pair states.

Table 49 TimeFinder/Clone operations for pair states - create action

TimeFinder pair state	STD copy source	STD copy target	BCV copy source	BCV copy target
Sync In Progress	Y	N	N	N
Synchronized	Y	Y	N	N
Restored	Y	Y	N	N

Pair states ruling TimeFinder/Snap operations

You can perform any TimeFinder/Snap action while the device(s) are in any of the following states:

- Split
- Split No Incremental
- Split Before Sync
- Split Before Restore

Create action

Table 50 identifies which `symsnap` copy **create** actions are available for use for each of the TimeFinder pair states.

Table 50 TimeFinder/Snap operations for pair states - create action

TimeFinder pair state	STD source[a]	BCV source[a]
Sync In Progress	Y	N
Synchronized	Y	N
Split In Progress	N	N
Restore In Progress	N	N
Restored	Y	N

a. The target device is always a VDEV.

Restore action

Table 51 identifies which `symsnap` copy **restore** actions are available for use for each of the TimeFinder pair states.

Table 51 TimeFinder/Snap operations for pair states - restore action

TimeFinder pair state	STD target[a]	BCV target[a]
Sync In Progress	N	N
Synchronized	Y	N
Split In Progress	N	N
Restore In Progress	N	N
Restored	Y	N

a. The source device is always a VDEV.

TimeFinder state rules for SRDF operations

Table 52 identifies which SRDF operations are available with various TimeFinder actions and states.

Table 52 TimeFinder state rules for SRDF operations

			SRDF Operations				
TimeFinder Action	**State**	**Member**	**Async Ready**	**Async Not Ready**	**Async Transmit Idle**	**Sync Ready**	**Sync Not Ready**
Establish	Suspend = True	R1 BCV	Cannot suspend (NAA)	Ok	Cannot suspend (ATI)	Ok (suspends)	Ok
		R2 BCV	Double writes (NAA)		Double writes (NAA)	Double writes (NAA)	
	Suspend = False	R1 BCV	Cannot suspend (NAA)		Cannot suspend (ATI)	Cannot suspend (IRS)	
		R2 BCV	Double writes (NAA)		Double writes (NAA)	Double writes (IRS)	
Split	No -remote	R1 BCV	Invalid state (IRS)	Ok	Invalid state (IRS)	Invalid state (IRS)	Ok
		R2 BCV					
Split -remote	Suspend = True	R1 BCV	Invalid state (IRS)	Cannot resume (NAA)	Invalid state (IRS)	Invalid state (IRS)	Ok (resumes)
		R2 BCV					Cannot resume (IRS)
	Suspend = False	R1 BCV					Cannot resume (FNA)
		R2 BCV					Cannot resume (FNA)
Restore	No -remote	R1 STD	Unintended copy (IRS)	Ok	Unintended copy (IRS)	Unintended copy (IRS)	Ok
		R2 STD	Double writes (IRS)		Double writes (IRS)	Double writes (IRS)	
Restore -remote	Suspend = True	R1 STD	Ok	Cannot resume (IRS)	Ok	Ok	Ok (resumes)
		R2 STD	Double writes (IRS)		Double writes (IRS)	Double writes (IRS)	Double writes (IRS)
	Suspend = False	R1 STD	Ok		Ok	Ok	Cannot resume (IRS)
		R2 STD	Double writes (IRS)		Double writes (IRS)	Double writes (IRS)	
Restore	Suspend = True	R1 BCV	Cannot suspend (NAA)	Ok	Cannot suspend (ATI)	Ok	Ok
		R2 BCV					
	Suspend = False	R1 BCV				Cannot suspend (IRS)	
		R2 BCV					

Index

Numerics

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R