

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

PRODUCT GUIDE
P/N 300-000-876
REV A06

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

Copyright © 2003, 2004, 2005 EMC Corporation. All Rights Reserved.

Printed January, 2005

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

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

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

Trademark Information

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

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

Chapter 1 Overview

Introduction to TimeFinder	1-2
TimeFinder Device Types	1-3
BCV Devices	1-3
Virtual Devices	1-3
SAVE Devices	1-4
Command Summary	1-6

Chapter 2 Performing TimeFinder Mirror Operations

The Business Continuanace Model	2-2
Standard Devices	2-2
BCV Devices	2-2
Business Continuanace	2-2
SRDF-Connected Symmetrix Sites	2-4
TimeFinder Operations Overview	2-8
Device External Locks	2-8
Disallow Synchronization Actions	2-9
TimeFinder/Mirror Control Operations	2-10
TimeFinder/Clone Emulation	2-12
Listing BCV Devices	2-14
List Host-Visible BCV Devices	2-14
List All BCV Devices	2-14
List BCV Devices with SCSI Reservations	2-14
Associating BCV Devices with a Device Group	2-15
Associating a BCV Device with a Device Group	2-15
Associating All Local BCV Devices	2-16

Associating All Remote BCV Devices	2-16
Associating an SRDF-Connected BCV Device	2-17
Associating an SRDF-Connected BCV Mirror Device	2-18
Associating Devices in a Compounded Remote Configuration	2-18
Disassociating BCV Devices from a Device Group	2-21
Disassociating an SRDF-Connected BCV Pair	2-21
Disassociating a Remote SRDF-Connected BCV Pair	2-22
Disassociating a Remote BCV of a Remote BCV Pair	2-22
Moving Devices	2-23
Moving a Specific Device	2-23
Moving All BCV Devices.....	2-23
Moving Remote BCV Devices	2-23
Removing Devices.....	2-24
Managing BCV Devices with Composite Groups.....	2-25
Associating BCV Devices with a Composite Group	2-25
Disassociating BCV Devices from a Composite Group	2-26
Moving BCV Devices to a Composite Group.....	2-26
Fully Establishing BCV Pairs	2-27
Establishing SRDF-Connected BCV Pairs.....	2-33
Establish Second-Level Remote BCV Pairs.....	2-35
Establishing Concurrent BCV Pairs	2-36
Protected BCV Establish (Moving All Mirrors)	2-38
Pairing Algorithm	2-39
Incrementally Establishing BCV Pairs.....	2-40
Establishing Multiple BCV Pairs.....	2-42
Incrementally Establishing SRDF-Connected BCV Pairs ..	2-43
Protected BCV Incremental Establish (Moving Mirrors) ..	2-46
Splitting BCV Pairs	2-47
Splitting SRDF-Connected BCV Pairs	2-50
Differential Split.....	2-53
Instant Split.....	2-54
Reverse Split.....	2-56
Splitting Concurrent BCV Pairs.....	2-58
Splitting a Protectively Established BCV	2-60
BCV Splits Impacting Database Applications	2-61
TimeFinder Consistent Split	2-64
Fully Restoring BCV Pairs	2-72
Restoring SRDF-Connected BCV Pairs	2-75
Incrementally Restoring BCV Pairs.....	2-79
Restoring (Incremental) Multiple BCV Pairs.....	2-82
Incremental Restore SRDF-Connected BCV Pairs.....	2-83

Protecting BCV Data During Full or Incremental Restores	2-86
Preferred Attachment of BCVs (Optional Operations).....	2-87
Attaching Remote Devices as Preferred Pairs	2-89
Detaching BCV Preferences from Devices.....	2-91
Detaching BCV Preferences for Remote Devices	2-92
Script Summary for Typical TimeFinder Operations	2-94
Script Example for Multi-BCV Environment	2-95
Querying BCV Pairs.....	2-96
Verifying BCV Pair States	2-97
BCV Pair States	2-99
Transient BCV Pair States	2-100
BCV Actions and Applicable States	2-100
symmir -g Command Line Options	2-101
Control Options.....	2-103
Verify Options.....	2-111
Query Options	2-112
symmir -cg Command Line Options.....	2-113
Remote Optimizing Option	2-115
Other Options	2-115
symmir -file Command Line Options	2-116
Various Remote Multi-Hop Configurations.....	2-119
System-Wide Device Groups.....	2-119
System-Wide Splits	2-120
Commands to Various Multi-Hop Devices and Links ...	2-122
Second-Level Controls for Multi-Hop SRDF	
Environments	2-124
Using the -remote Option on Multi-Hop Split Actions..	2-125
Relabeling Devices for Windows 2000	2-127

Chapter 3 Performing TimeFinder/Clone Operations

TimeFinder/Clone Environment	3-2
Clone Operations Overview	3-2
Pair States Ruling Clone Operations.....	3-3
Creating a Clone Copy Session	3-5
Copy Session Limits	3-5
CopyOnAccess	3-6
Full Device Copy Option	3-6
Precopy Option	3-6
Differential Option.....	3-7
Copy Functionality by Target Type	3-7
Establish Command.....	3-8
Modifying a Clone Copy Session.....	3-9

Activating a Clone Copy Session	3-10
Not Ready Option	3-10
Consistent Option.....	3-10
Recreating a Clone Copy Session	3-12
Precopy Option.....	3-12
Establish Command	3-12
Restoring Data from a Target Device	3-13
Terminating a Clone Copy Session	3-14
BCV as the Clone Source	3-15
Creating Multiple Clone Copies.....	3-18
Standard Device Clone Copies	3-18
BCV Device Clone Copies	3-20
symclone -g Command Line Options.....	3-23
symclone -file Command Line Options.....	3-25
symclone -cg Command Line Options	3-27

Chapter 4 **Performing TimeFinder/Snap Operations**

TimeFinder/Snap Environment	4-2
Snap Operations Overview	4-3
Pair States Ruling Snap Operations	4-4
Creating a Virtual Copy Session.....	4-5
Copy Session Limits	4-5
Copy On First Write	4-6
Specifying a SAVE Device Pool	4-7
Monitoring SAVE Device Usage.....	4-7
Activating a Virtual Copy Session.....	4-9
Not Ready Option	4-9
PowerPath Option.....	4-9
Consistent Option.....	4-10
Terminating a Virtual Copy Session.....	4-12
Using a BCV as the Snap Source	4-13
Creating Multiple Virtual Copies	4-16
Standard Source Device Copies.....	4-16
BCV Source Device Virtual Copies	4-18
Attaching Source and Target Virtual Devices.....	4-20
Restoring Data from Virtual Devices	4-21
Persistent Restore	4-22
Incremental Restore to Source	4-22
Incremental Restore to a BCV	4-23
Full Restore to Anywhere.....	4-27
symsnap -g and -cg Command Line Options.....	4-29
symsnap -file Command Line Options	4-31

Appendix A SRDF State Rules Reference

SRDF Pair States	A-2
State Rules for TimeFinder/Clone Operations	A-4
Sync In Progress	A-4
Synchronized	A-4
Split	A-5
Failed Over.....	A-5
R1 Updated	A-6
R1 Update In Progress.....	A-6
Suspended.....	A-7
Partitioned.....	A-7
Consistent.....	A-8
Invalid.....	A-8
State Rules for TimeFinder/Snap Operations	A-9
Sync In Progress	A-9
Synchronized	A-10
Split	A-10
Failed Over.....	A-11
R1 Updated	A-11
R1 Update In Progress.....	A-12
Suspended.....	A-12
Partitioned.....	A-13
Invalid.....	A-13
Consistent.....	A-14

Appendix B TimeFinder State Reference

TimeFinder Pair States	B-2
Pair States Ruling TimeFinder/Clone Operations	B-3
Pair States Ruling TimeFinder/Snap Operations	B-4

Index	i-1
-------------	-----

Figures

2-1	A Simple TimeFinder Configuration	2-3
2-2	SRDF: Mirroring the Local Standard	2-4
2-3	SRDF: Mirroring the Local BCVs	2-5
2-4	SRDF: Mirroring the Remote BCVs	2-6
2-5	Mirror Configuration Types	2-7
2-6	Compounded Remote Configuration Example	2-20
2-7	Initial BCV Configuration	2-27
2-8	Fully Establishing a BCV Pair	2-29
2-9	Establishing a Multi-BCV Environment	2-31
2-10	Canceling a Multi-BCV	2-32
2-11	Establish the Remote BCV Pair (Mirroring the Local STD)	2-33
2-12	Establish the Remote BCV Pair (Mirroring the Local BCV)	2-34
2-13	Establish the Remote Hop 2 BCV Pair (Mirroring the Local BCV)	2-35
2-14	Establish Concurrent BCV Pairs	2-36
2-15	Establish Two-Way BCV Mirrors with Protected Establish	2-38
2-16	Incremental Establish of a BCV Pair	2-41
2-17	Establish a Multi-BCV Environment	2-42
2-18	Incremental Establish of the Remote BCV to Mirror Local STD	2-44
2-19	Incremental Establish of the Remote BCV to Mirror Local BCV	2-45
2-20	Split the BCV Pair	2-48
2-21	Split the Remote BCV Pair That Mirrors the Local STD	2-50
2-22	Split the Remote BCV Pair That Mirrors the Local BCV	2-51
2-23	Remote Copy with a BCV Split	2-52
2-24	Differential Copy in a Differential Split	2-53
2-25	Two-Way Mirror BCV Establish/Split Normal Behavior	2-56
2-26	2-Way Mirror Reverse Establish/Split Behavior	2-57
2-27	Splitting Concurrent BCV Pairs	2-58
2-28	Split Behavior on Two-Way BCV Mirrors	2-60
2-29	An Established Oracle DB Environment	2-63

2-30	Consistency Split on Oracle Environment	2-66
2-31	Consistent Split on Both RDF Sides	2-67
2-32	ECA Consistent Split	2-68
2-33	Full Restore of the BCV Pair	2-73
2-34	Full Restore of Remote STD That Mirrors Local STD	2-75
2-35	Restore of Remote STD That Mirrors Local BCV	2-76
2-36	Remote Copy with Restore	2-77
2-37	Restoring with the Remote Host	2-78
2-38	Incremental Restore the STD	2-80
2-39	Restoring a BCV in a Multi-BCV Environment	2-82
2-40	Incremental Restore the Remote STD that Mirrors Local STD	2-84
2-41	Incremental Restore of the Remote STD that Mirrors Local BCV	2-85
2-42	Preferred Attachment for BCV Pairing	2-88
2-43	Preferred Attachment of Remote BCV to Mirror Local STD	2-89
2-44	Preferred Attachment of Remote BCV to Mirror Local BCV	2-90
2-45	Detach a Preferred BCV Attachment	2-91
2-46	Detach a BCV Preference from Remote STD (Mirrored Local STD) ...	2-92
2-47	Detach a BCV Preference from Remote STD (Mirrored Local BCV) ...	2-93
2-48	Various Remote Configurations	2-121
2-49	Control Operations on Multi-Hop SRDF Configurations	2-123
2-50	The -remote Option on Multi-Hop Configurations	2-126
3-1	TimeFinder/Clone Copy of a Standard Device	3-3
3-2	TimeFinder/Clone Consistent Activate Using ECA	3-11
3-3	Creating a TimeFinder/Clone from a BCV Source	3-16
3-4	Creating Multiple Standard Device Clone Copies	3-18
3-5	Creating Multiple BCV Device Clone Copies	3-20
4-1	Copy of a Standard Device to a Virtual Device (VDEV)	4-3
4-2	TF/Snap Consistent Activate Using ECA	4-11
4-3	Creating a Virtual Copy from a BCV	4-14
4-4	Creating Multiple Standard Device Virtual Copies	4-16
4-5	Creating Multiple Virtual Copies from a BCV Device	4-18
4-6	Incremental Restore to a BCV	4-25

Tables

1-1	TimeFinder Command Summary	1-6
2-1	TimeFinder Commands Mapped to Clone Operation	2-12
2-2	Consistent Split Comparison: ECA Versus PowerPath	2-70
2-3	BCV Pair States	2-99
2-4	Actions for BCV Devices	2-100
2-5	BCV Control Actions and Applicable States	2-100
2-6	symmir -g Control Arguments and Possible Options	2-101
2-7	symmir -g View Arguments and Possible Options	2-102
2-8	Results of -force Option	2-106
2-9	RDF Pair States Relating to the Remote Option	2-109
2-10	symmir -cg Control Arguments and Possible Options	2-113
2-11	symmir -cg View Arguments and Possible Options	2-114
2-12	symmir -file Control Arguments and Possible Options	2-116
2-13	symmir -file View Arguments and Possible Options	2-117
3-1	Comparing Target Type	3-7
3-2	symclone -g Control Arguments and Possible Options	3-23
3-3	symclone -file Control Arguments and Possible Options	3-25
3-4	symclone -cg Control Arguments and Possible Options	3-27
4-1	symsnap -g and -cg Control Arguments and Possible Options	4-29
4-2	symsnap -file Control Arguments and Possible Options	4-31
A-1	SRDF Pair States	A-2
A-2	TimeFinder/Clone Operations for SyncInProgress Pair States	A-4
A-3	TimeFinder/Clone Operations for Synchronized Pair States	A-4
A-4	TimeFinder/Clone Operations for Split Pair States	A-5
A-5	TimeFinder/Clone Operations for Failed Over Pair States	A-5
A-6	TimeFinder/Clone Operations for R1 Updated Pair States	A-6
A-7	TimeFinder/Clone Operations for R1 UpdInProgress Pair States	A-6
A-8	TimeFinder/Clone Operations for Suspended Pair States	A-7
A-9	TimeFinder/Clone Operations for Partitioned Pair States	A-7

A-10	TimeFinder/Clone Operations for Consistent Pair States	A-8
A-11	TimeFinder/Clone Operations for Invalid Pair States	A-8
A-12	TimeFinder/Snap Operations for SyncInProgress Pair States	A-9
A-13	TimeFinder/Snap Operations for Synchronized Pair States	A-10
A-14	TimeFinder/Snap Operations for Split Pair States	A-10
A-15	TimeFinder/Snap Operations for Failed Over Pair States	A-11
A-16	TimeFinder/Snap Operations for R1 Updated Pair States	A-11
A-17	TimeFinder/Snap Operations for R1 UpdInProgress Pair States	A-12
A-18	TimeFinder/Snap Operations for Suspended Pair States	A-12
A-19	TimeFinder/Snap Operations for Partitioned Pair States	A-13
A-20	TimeFinder/Snap Operations for Invalid Pair States	A-13
A-21	TimeFinder/Snap Operations for Consistent Pair States	A-14
B-1	TimeFinder Pair States	B-2
B-2	TimeFinder/Clone Operations for Pair States - CreateAction	B-3
B-3	TimeFinder Snap Operations for Pair States - Create Action	B-4
B-4	TimeFinder Snap Operations for Pair States - RestoreAction	B-4

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

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

Audience This manual provides both guide and reference information for command-line users and script programmers that describes how to manage BCV devices for a TimeFinder environment using the SYMCLI commands of the EMC Solutions Enabler software.

Organization The following defines the structure of this manual:

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

Chapter 2, *Performing TimeFinder Mirror Operations*, describes how to perform BCV control and management operations using the `symmir` SYMCLI command.

Chapter 3, *Performing TimeFinder/Clone Operations*, describes how to create and manage Clone copies using the `symclone` SYMCLI command.

Chapter 4, *Performing TimeFinder/Snap Operations*, describes how to perform Snap control operations on devices using the `symsnap` SYMCLI command.

Appendix A, *SRDF State Rules Reference*, describes the applicable SRDF pair states that rule TimeFinder/Clone and TimeFinder/Snap control operations within each state.

Appendix B, *TimeFinder State Reference* describes the applicable TimeFinder pair states that rule TimeFinder/Clone and TimeFinder/Snap copy session operations.

Detailed man page descriptions of all SYMCLI commands, environment variables, option file parameters, and error codes can now be found in the companion *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Related Documentation

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

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

The following are EMC Engineering white papers of related interest:

- ◆ *Using SYMCLI to Set Up TimeFinder/Mirror BCV Pairs*
- ◆ *Using SYMCLI to Attach, Query, and Verify With TimeFinder Family Products*
- ◆ *Using SYMCLI to Perform TimeFinder/Mirror Control Operations*
- ◆ *Using SYMCLI to Set Up TimeFinder/Mirror Multiple BCVs and Concurrent BCVs*
- ◆ *Using SYMCLI to Perform Consistent Splits With TimeFinder Product Family*
- ◆ *Using SYMCLI to Perform TimeFinder/Mirror Clone Operations*
- ◆ *Using SYMCLI to Perform TimeFinder/Mirror Snapshot Operations*

- ◆ *Instant Restore of Oracle Databases: Leveraging TimeFinder Multi-BCV and Instant Split*

Conventions Used in This Manual

The following conventions are used in this manual:

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

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



CAUTION

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

Typographical Conventions

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

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

Where to Get Help

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

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

United States: (800) 782-4362 (SVC-4EMC)

Canada: (800) 543-4782 (543-4SVC)

Worldwide: (508) 497-7901

Language services are available upon request.

Sales and Customer Service Contacts

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

<http://emc.com/contact/>

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

<http://powerlink.emc.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send a message to **techpub_comments@emc.com** with your opinions of this guide.

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

Overview

This chapter introduces the Solutions Enabler TimeFinder component and describes the types of devices used in TimeFinder configurations.

- ◆ Introduction to TimeFinder 1-2
- ◆ TimeFinder Device Types..... 1-3
- ◆ Command Summary 1-6

Introduction to TimeFinder

The SYMCLI TimeFinder® component extends the basic SYMCLI command set to include TimeFinder or business continuance commands that allow you to perform control operations on BCV device pairs within your TimeFinder environment. This manual specifically describes the functionality of:

- ◆ TimeFinder/Mirror — General monitor and control operations
- ◆ TimeFinder/Snap — Snap copy sessions
- ◆ TimeFinder/Clone — Clone copy sessions
- ◆ TimeFinder/CG — Consistency groups

Commands such as `symmir`, `symbcv`, and `symioctl` allow you to perform a wide spectrum of monitor and control operations on standard/BCV device pairs within your TimeFinder/Mirror environment. With the use of the `symioctl` command, you can send I/O control commands to a specified database server.

The TimeFinder/Clone command, `symclone`, allows you to create point-in-time copies of a source device on multiple target devices. The TimeFinder/Snap command, `symsnap`, allows you to create virtual device copy sessions between a source device and multiple *virtual* (VDEV) target devices.

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

TimeFinder Device Types

This section describes the device types that are specific to TimeFinder configurations.

BCV Devices

Symmetrix TimeFinder is a business continuance solution that allows you to use special Symmetrix devices called *business continuance volumes* (BCVs). A *BCV device* contains a copy of data from a standard Symmetrix device that is online for regular I/O operation from its host(s). Uses for the BCV copies can include backup, restore, decision support, and applications testing. Each BCV device has its own host address, and is configured as a stand-alone Symmetrix device.

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

Virtual Devices

TimeFinder/Snap operations provide instant snap device copies, using *virtual devices*. A virtual device is a host-accessible device containing track-level location information (pointers), that indicates where the copy session data is located in the physical storage. Virtual devices consume minimal physical disk storage, as they store only the address pointers to the data stored on the source device or 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 via the corresponding virtual device. Upon a first write to the source device during the copy session, a pre-updated image of the changed track will be 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.

When there is a host write during the virtual copy session, that data is immediately copied to the SAVE device (refer to *SAVE Devices* later on this page).

The attached host views the point-in-time copy via the 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* in Chapter 4.

For more information on how virtual devices are implemented in a copy session, refer to Chapter 4, *Performing TimeFinder/Snap Operations*.

Virtual and SAVE devices must be configured on the Symmetrix array by using the Symmetrix Configuration Manager software. For information on how to configure devices using the SYMCLI, refer to the *EMC Solutions Enabler Symmetrix Configuration Change CLI Product Guide*.

SAVE Devices

Symmetrix *SAVE devices* are specially configured devices (not mapped to the host) that provide pooled physical storage space used to store pre-update images or changed tracks during a virtual copy session. Devices within a SAVE device pool act as a group for storing data in striped form. Symmetrix supports the creation of multiple named SAVE device pools, allowing `symsnap` commands to use a particular pool.

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

Note: Configuring SAVE devices as unprotected is not recommended and requires special approval by EMC.

SAVE devices are configured with a limited amount of available storage space and require careful planning before use. 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 and the session will fail. However, SAVE devices can be monitored for the percentage full and can be expanded. For more information on how SAVE devices are configured, refer to the *EMC Solutions Enabler Symmetrix Configuration Change CLI Product Guide*.

For more information on how SAVE devices are implemented in a copy session, refer to *Creating a Virtual Copy Session* on page 4-5.

Command Summary

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, or TimeFinder device groups. Table 1-1 lists and summarizes these commands. For more detail about the command syntactical form, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Table 1-1 TimeFinder Command Summary

Command	Description
<code>symbcv</code>	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.
<code>symclone</code>	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 sixteen target devices. • 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 copied state for devices in a device group. • Lists all copy sessions that have been created on the Symmetrix array.

Table 1-1 TimeFinder Command Summary (continued)

Command	Description
<code>symmir</code>	Performs control operations on BCV device pairs including: - Establishes (mirror) one or all standard devices in a device group with one or more BCV devices that are associated locally or remotely with the group. The operation can be a full or incremental establish. - Restores one or all standard devices in a device group from one or more BCV devices that are associated locally or remotely with the group. The operation can be a full or incremental restore. - Splits one or all BCV devices from the standard device(s) in a device group. - Returns information about the state of mirroring of one or all BCV device pairs in a device group.
<code>symsnap</code>	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 sixteen target devices. - 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. - 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 the CopyOnWrite state for devices in a device group. - Attaches and detaches target devices as the preferred devices to use in a requested Snap operation. - Restores a virtual device to another 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 1-1 TimeFinder Command Summary (continued)

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

Performing TimeFinder Mirror Operations

This chapter describes the business continuance model and how to manage and control TimeFinder/Mirror (BCV) devices using SYMCLI.

◆ The Business Continuance Model	2-2
◆ TimeFinder Operations Overview	2-8
◆ Listing BCV Devices	2-14
◆ Associating BCV Devices with a Device Group	2-15
◆ Disassociating BCV Devices from a Device Group	2-21
◆ Moving Devices	2-23
◆ Managing BCV Devices with Composite Groups	2-25
◆ Fully Establishing BCV Pairs	2-27
◆ Incrementally Establishing BCV Pairs	2-40
◆ Splitting BCV Pairs	2-47
◆ Fully Restoring BCV Pairs	2-72
◆ Incrementally Restoring BCV Pairs	2-79
◆ Protecting BCV Data During Full or Incremental Restores	2-86
◆ Preferred Attachment of BCVs (Optional Operations)	2-87
◆ Script Summary for Typical TimeFinder Operations	2-94
◆ Script Example for Multi-BCV Environment	2-95
◆ Querying BCV Pairs	2-96
◆ BCV Pair States	2-99
◆ symmir -g Command Line Options	2-101
◆ symmir -cg Command Line Options	2-113
◆ symmir -file Command Line Options	2-116
◆ Various Remote Multi-Hop Configurations	2-119
◆ Relabeling Devices for Windows 2000	2-127

The Business Continuance Model

Symmetrix TimeFinder is essentially a business continuance solution that allows you to use 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.

Standard Devices

Standard devices are Symmetrix disks that are configured for normal Symmetrix operations under a desired protection method (such as RAID 1, RAID-S, RAID 5, and SRDF).

The standard device can have any mirror structure (unprotected, one mirror, two mirrors), 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.

Devices that are *meta head* devices can be standard devices. *Meta members* are implicitly controlled when the meta head is controlled.

BCV Devices

A *BCV device* is a standard Symmetrix device with special attributes that allow it to be accessed independently of its associated standard device to support host applications and processes. A BCV device can have a mirror, remote mirror (R2), or spare while it is not mirrored to a standard device.

Meta head devices can also be BCV devices, and the meta members will be implicitly controlled when the BCV meta head is manipulated.

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 via its original device address while it is in an established pair. Once the BCV device is synchronized, it may be separated (*split*) later from the standard device with which it is paired. The BCV device with the synchronized data becomes available for backup or other host processes through its original device address.

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.

One or several hosts can be attached to a Symmetrix array containing the BCV devices (Figure 2-1). Any Symmetrix system, including those configured for protection modes, such as RAID (1 or S) and SRDF, support the BCV option.

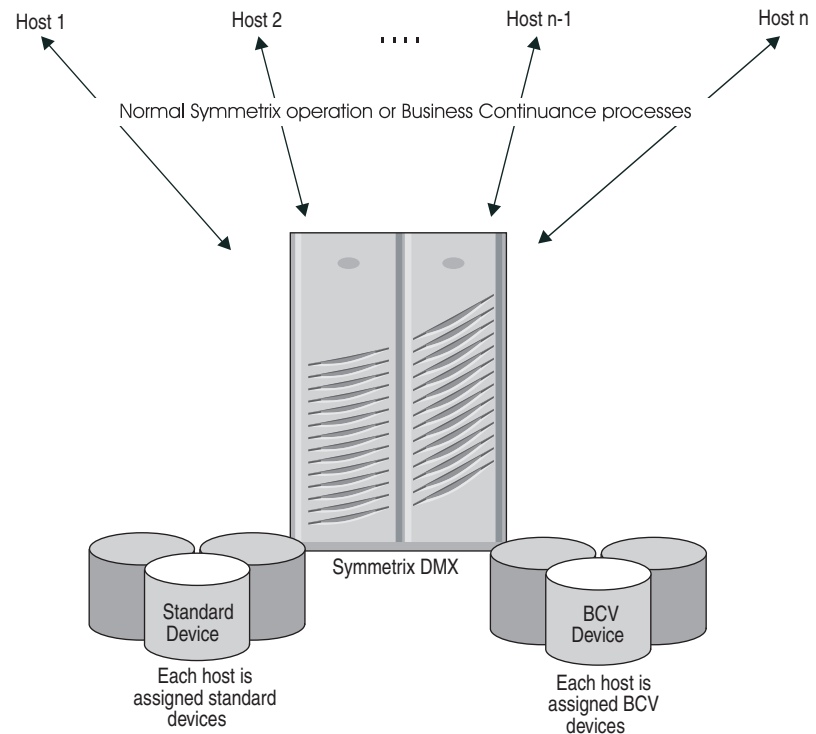


Figure 2-1 A Simple TimeFinder Configuration

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 that remote operations with SRDF-connected BCVs are also described in *Associating an SRDF-Connected BCV Device* on page 2-17 and in the *EMC Solutions Enabler Symmetrix TimeFinder CLI Product Guide*.

Remotely Mirroring the Local Standard

There are multiple types of SRDF-connected BCV devices. An SRDF-connected BCV can be paired with the R2 mirror of the local RDF standard devices (RBCV) as shown in Figure 2-2.

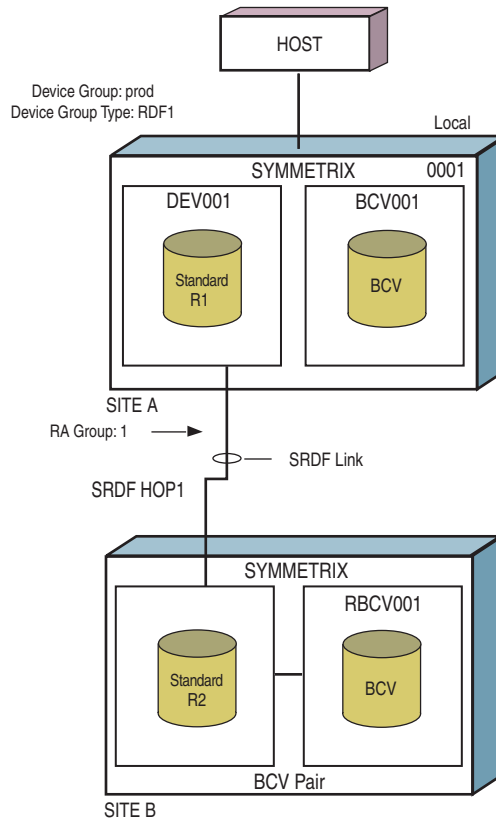


Figure 2-2 SRDF: Mirroring the Local Standard

Remotely Mirroring the Local BCVs

In another scenario, the SRDF-connected BCV can be paired with the 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 2-3).

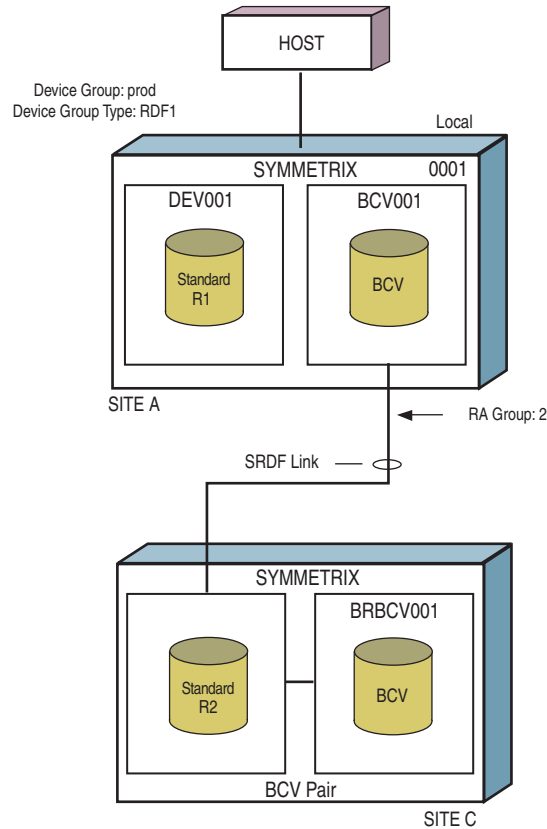


Figure 2-3 SRDF: Mirroring the Local BCVs

Multi-Hop Mirroring the Remote BCVs

In a multi-hop configuration, the SRDF-connected BCV can be paired with the 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 2-4).

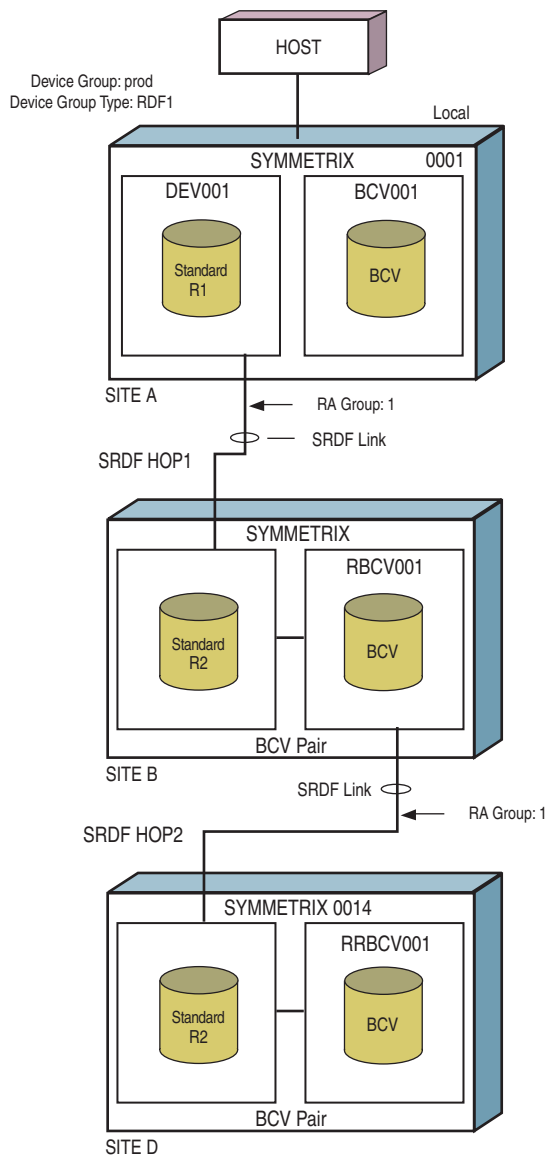


Figure 2-4 SRDF: Mirroring the Remote BCVs

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.

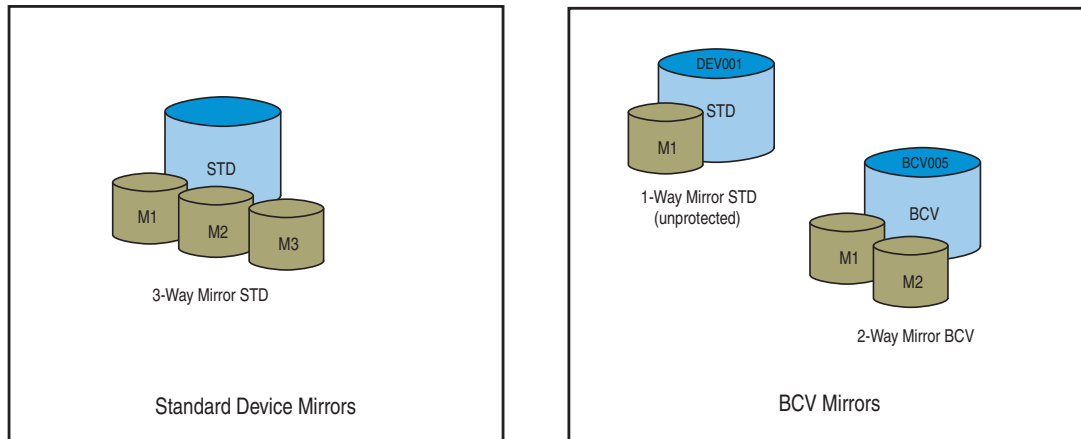


Figure 2-5 Mirror Configuration Types

TimeFinder Operations Overview

The TimeFinder/Mirror control operations described in this chapter make use of BCV devices that were previously associated with a device group or composite group using the `symbcv` command, which provides a foundation for various host Business Continuity (BC) management operations. TimeFinder/Mirror control functionality described in this chapter requires 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 Continuity 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.

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

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

On your host, if you discover a lock that has been on for well over 2 hours and are sure no one is using the device resources, you can choose to release the lock.

To release the device lock on a range of Symmetrix devices in Symmetrix 870, enter:

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



CAUTION

Use the release lock action only if you believe that the Symmetrix device lock was forgotten and there are NO other operations in progress to the specified Symmetrix devices (local or remote). Locks are typically short duration (one second to an hour or so). But, be ready to recognize when a device lock is being held or preserved by a certain application (such as an SRDF or TimeFinder/Snap action), which might be allocated as a long-duration lock.

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.

Note: The skip locks option is also available during TimeFinder/Clone (`symclone`) and TimeFinder/Snap (`symsnap`) operations when used with the `create`, `activate`, or `terminate` commands.

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

Group Operations

With Solutions Enabler V5.4, TimeFinder operations using the SYMCLI `symmir`, `symclone`, and `symsnap` commands have been extended to support composite groups (`-cg`) or devices in a composite group, as well as device groups (`-g`) and devices within a device group. For information about composite groups and device groups, refer to the *Solutions Enabler Symmetrix Base Management Product Guide*.

TimeFinder/Mirror Control Operations

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

- ◆ Associate a device pair.
Associate a BCV device (from the specified device group) with the specified standard device as a prospective pair.
- ◆ Disassociate a device pair.
Disassociate the associated BCV device from the specified standard device of this existing pair.
- ◆ List all BCV devices in the Symmetrix array.
- ◆ Move a BCV device from one group to another.
- ◆ Remove all BCV devices from the specified device group.

The `symmir` command allows 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 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 two 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.

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 supports the ability to map TimeFinder commands to Clone operations. With this feature, TimeFinder commands will execute Clone operations.

This feature is enabled by setting the environment variable SYMCLI_CLONE_EMULATION to ENABLED, or if a RAID 5 BCV is encountered.

Table 2-1 details how TimeFinder commands map to Clone operations.

Table 2-1 TimeFinder Commands Mapped to Clone Operation

TimeFinder symmir Command	Clone (symclone) Operations
FULL ESTABLISH	CREATE with Pre-copy
SPLIT	ACTIVATE
INCREMENTAL ESTABLISH	RECREATE
FULL RESTORE	FULL RESTORE
INCREMENTAL RESTORE	INCREMENTAL RESTORE
SymCLI VERIFY	SymCLI VERIFY
SymCLI ATTACH	SymCLI ATTACH
SymCLI DETACH	SymCLI DETACH
CANCEL	TERMINATE
PROTECT RESTORE	Default feature
PROT BCV ESTAB	Default feature ^a
SymCLI QUERY	SymCLI QUERY
BG SPLIT	COPYINPROG

a. Only after the completion of split is the target device fully synchronized as in a protected BCV Establish.

The following limitations apply to this feature:

- ◆ 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.
- ◆ The SYMAPI_DEFAULT_BCV_SPLIT_TYPE, SYMAPI_DEFAULT_BCV_RESTORE_TYPE, SYMAPI_DEFAULT_BCV_ESTABLISH_TYPE option file settings will be ignored.
- ◆ Any Snap or Clone operation to an emulated device will be rejected.
- ◆ The maximum number of BCVs that can be incrementally established with a Standard device will be 8 instead of the 16 allowed by TimeFinder. SYMCLI_MAX_BCV_PAIRS can at most be eight.
- ◆ The BCV states Split-Before-Restore and Split-Before-Sync are not valid states for an emulation device. In both cases, a forced Split will complete the 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.

List Host-Visible BCV Devices

To list all the BCV devices that are visible to your host, enter:

```
symbcv list pd
```

List All BCV Devices

To list all the BCV devices, regardless of whether they are visible to your host, enter:

```
symbcv list dev
```

Note: The `dev` statement is optional. The truncated command `symbcv list` will return the same set of values.

List BCV Devices with SCSI Reservations

To list all the BCV devices that have SCSI reservations, regardless of whether they are visible to your host, enter:

```
symbcv -resv list
```

Associating BCV Devices with a Device Group

Various compound (even multi-hop) 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.

Note: For more information about device groups, refer to *BCV Devices* on page 2-2.

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. 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.

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 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 that are visible to your host with a device group `prod`, enter:

```
symbcv -g prod associateall
```

Associating All Remote BCV Devices

All remote BCV devices can be associated 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)

Must be used with the `-rdf` option.

- ◆ `-rrdf` specifies BCVs that are remotely associated with remote BCVs (RRBCVs)

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 2-6 on page 2-20 shows a device group of type RDF1 (there are two other device group types, RDF2 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. Any one BCV device cannot be associated with more than one group at the same time.

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 100005300021 -g prod\  
-rdf -RDFG 1 associate dev 000B RBCV001
```

where:

`100005300021` 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 2-6 on page 2-20.

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 2-6 on page 2-20.

For example:

```
symbcv -sid 100005300021 -g prod\  
-rdf -bcv -RDFG 2 associate dev 002A BRBCV001
```

where:

100005300021 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

As shown in Figure 2-6 on page 2-20, you can have a site using SRDF links to remotely mirror a local Symmetrix array at Site A. Remote Site B will function as the remote mirror to the standard devices at Site A. A third site, using an SRDF link (remote Site C), remotely mirrors just the BCV devices in the Symmetrix at Site A. A second-level SRDF can also be multi-hopped where Remote Site D 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 2-6 on page 2-20. 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 000C 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 of the remote BCV (RRBCV) device 000D is added to group `prod` and given a logical name of `RRBCV001`, by entering:

```
symbcv -g prod associate dev 000D -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`).

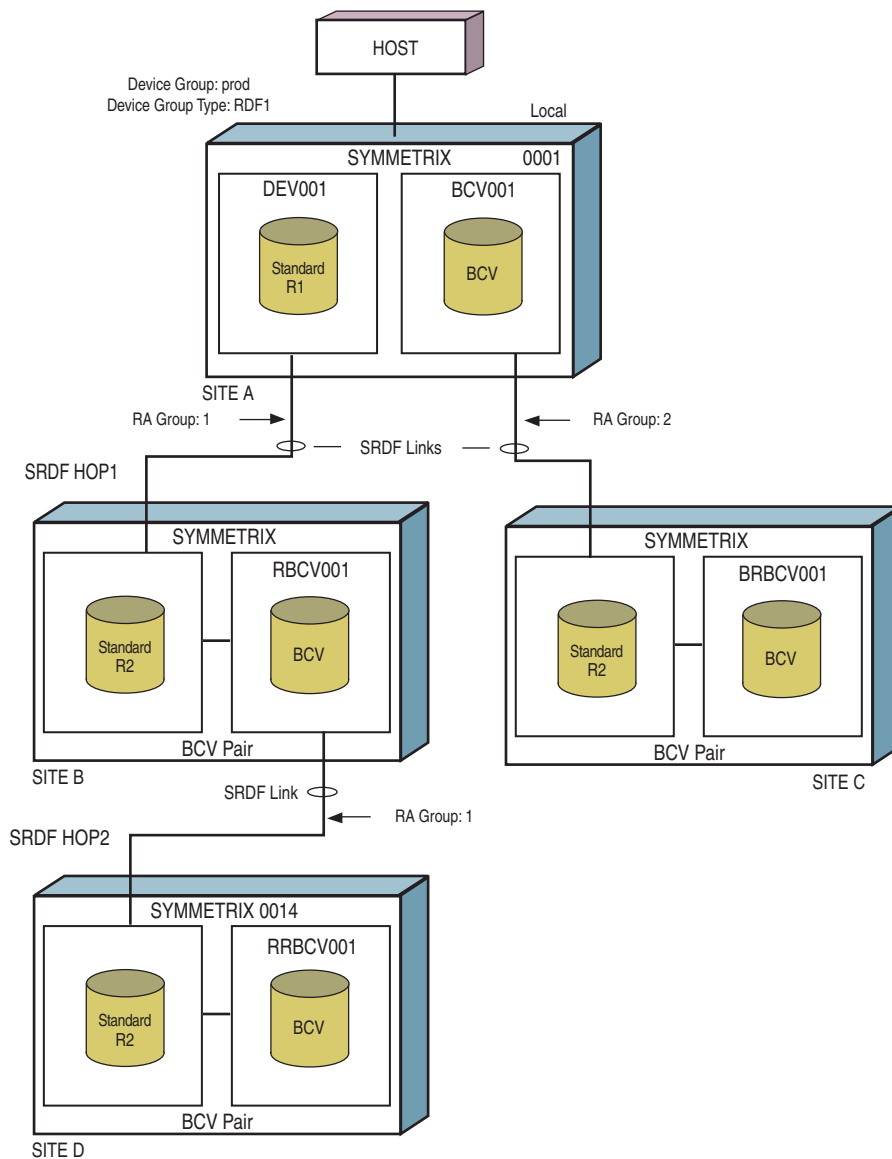


Figure 2-6 Compounded Remote Configuration Example

Disassociating BCV Devices from a Device Group

After you have associated a BCV device 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 with 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 also disassociate a remote BCV pair that mirrors the local BCV device, which is shown associated in Figure 2-6 on page 2-20.

For example:

```
symbcv -sid 100005300021 -g SAMPLE1\  
-rdf -bcv disassociate dev 002A
```

where:

100005300021 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 also disassociate a remote BCV that mirrors the remote BCV device, which is shown associated in Figure 2-6 on page 2-20. 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 100005300021 -g SAMPLE1\  
-rrdf disassociate dev 002A
```

where:

100005300021 is the ID of the local Symmetrix array

`-rrdf` specifies the BCV is being remotely associated with a remote BCV in the group

Moving Devices

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 `DEV001` from device group `prod` to the destination group `NewGroup` and rename the moved device, enter:

```
symld -g prod -sid 0001 move DEV001 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 associate 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)

Note: Must be used with the `-rdf` option.

- ◆ `-rrdf` specifies BCVs that are remotely associated with remote BCVs (RRBCVs)

Removing Devices

The `symbcv` command also contains a remove all action (`rma11`), 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 `-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 CG:

```
symbcv -sid SymmID -cg CgName [-rdf] [-bcv] [-RDFG
GrpNum] associate dev SymDevName
```

Use the following syntax to associate all BCV devices for the given *SymmID* to the CG:

```
symbcv -sid SymmID -cg CgName [[-rdf] [-bcv] | [-rrdf]]
[-RDFG GrpNum] [-SA <#|ALL>] [-P #] [-CAP #] [-N #]
[-R1|-NOR1] [-R2|-NOR2]
[-RANGE <StartDev>:<EndDev>] [-host HostName]
associateall [dev|-host HostName]
```

Disassociating BCV Devices from a Composite Group

Use the following syntax to disassociate a BCV using the given Symmetrix device name:

```
symbcv -sid SymmID -cg CgName [-force]
[[-rdf] [-bcv] | [-rrdf]] [-RDFG GrpNum]
disassociate dev SymDevName
```

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 CG:

```
symbcv -sid SymmID -cg CgName [-force]
[[-rdf] [-bcv] | [-rrdf]] [-RDFG GrpNum]
[-dest_sid DestSymmID] [-dest_RDFG DestGrpNum]
move dev SymDevName DestCgName
```

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 2-7 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.

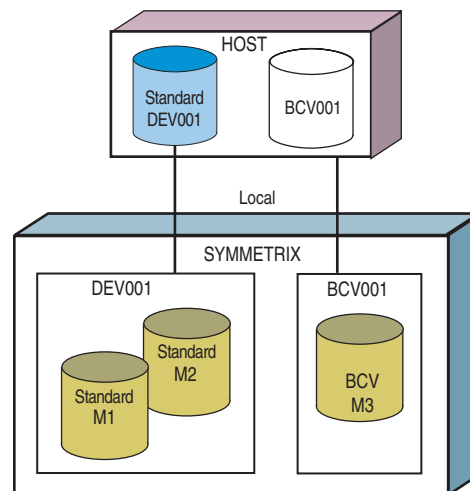


Figure 2-7 Initial BCV Configuration

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 -g CgName -full establish
symmir -f[ile] FileName -full establish
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

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 1d BCV005 \
    DEV002 BCV 1d BCV007 \
    DEV003 BCV 1d BCV008
```

Figure 2-8 illustrates the full establish of a BCV pair.

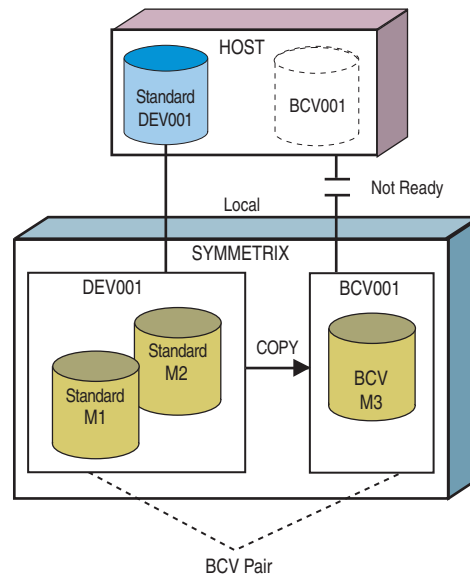


Figure 2-8 Fully Establishing a BCV Pair

Note: Devices must have been previously associated with the device group or composite group using the `symrdf -f FileName` or `symbcv` command.

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 may be the second, third, or fourth mirror of the standard device. For example, in Figure 2-8, it is the third mirror (M3).

- ◆ The contents of the standard device are copied to the BCV. For example, in Figure 2-8 on page 2-29, the BCV device receives its data from both the M1 and M2 devices.

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.

Pairing Algorithm

If no other overriding pairing options are specified, or if the pair was never previously established or attached, the standard and BCV device pairing algorithm selects the devices in a group such that they are of equal storage size for a given pair. For more information about pairing options (`-exact`, `-opt`, and `-opt_rag`), refer to Table 2-6 on page 2-101.

To use a BCV device for BC procedures, you must split the BCV pair to make the BCV device available to the host.

If a BCV device has its own local or remote mirrors, these mirrors will become synchronized after the BCV pair is split. While the BCV pair is synchronized, these mirrors remain not ready.

Establishing Multiple BCV Pairs

You can fully establish (at different times) up to eight BCV devices 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 2-9 on page 2-31), a multi-BCV environment becomes established that retains progressive historical snaps 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.

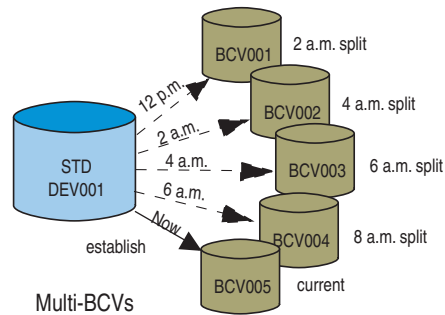


Figure 2-9 Establishing a Multi-BCV Environment

For example, to fully establish the standard with BCV005, enter:

```
symmir -g DgName split DEV001
symmir -g DgName -full establish DEV001 BCV 1d 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 *Restoring (Incremental) Multiple BCV Pairs* on page 2-82).

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.

For example, to remove BCV001 from the set, enter:

```
symmir -g DgName cancel DEV001 BCV 1d BCV001
```

As shown in Figure 2-10, once you cancel the specified BCV, all records of track changes between the STD and the canceled BCV are destroyed.

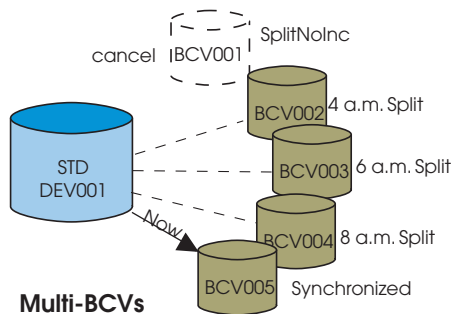


Figure 2-10 Canceling a Multi-BCV

Establishing SRDF-Connected BCV Pairs

You can also specify an establish action to a remote Symmetrix site using the RDF flag (`-rdf` option), which fully establishes the remote BCV pairs.

For example, Figure 2-11 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.

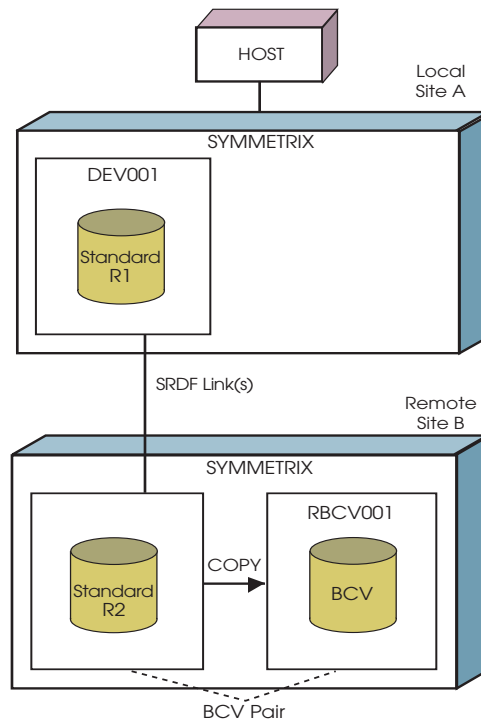


Figure 2-11 Establish the Remote BCV Pair (Mirroring the Local STD)

Figure 2-12 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 1d BRBCV001
```

In this case, the 1d parameter 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.

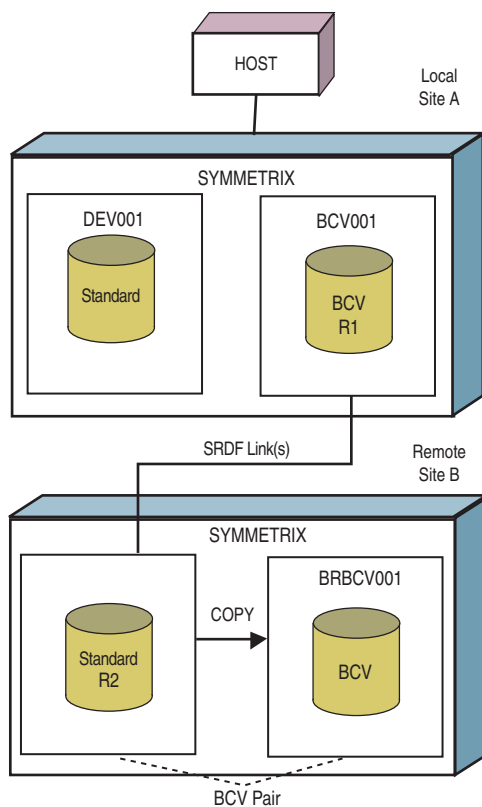


Figure 2-12 Establish the Remote BCV Pair (Mirroring the Local BCV)

Establish 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 2-13 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
```

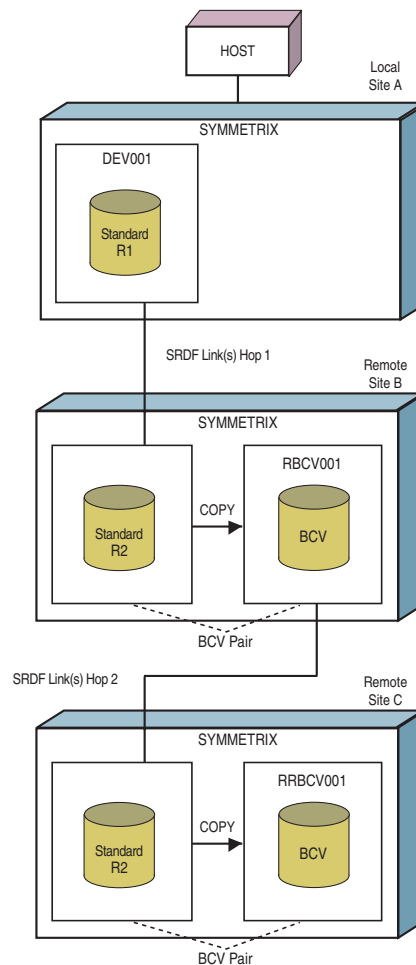


Figure 2-13 Establish the Remote Hop 2 BCV Pair (Mirroring the Local BCV)

To initiate a full establish on one remote BCV pair, RBCV001, in the `prod` group, enter:

```
symmir -g prod -rrbcv -full establish \
      RBCV001 BCV 1d 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 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 (Engenuity Version 5x66 minimum), that is known as a multi-BCV relationship.

Starting with the Engenuity Version 5568, you can establish two BCV devices 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 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 1d BCV001\
      DEV012 bcv 1d BCV002
```

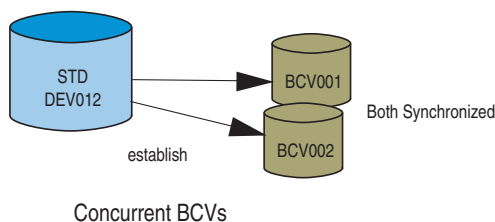


Figure 2-14 Establish Concurrent BCV Pairs

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 once BCV with it.

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.

Figure 2-15 compares the Symmetrix array behavior between a normal and a moving protected establish action.

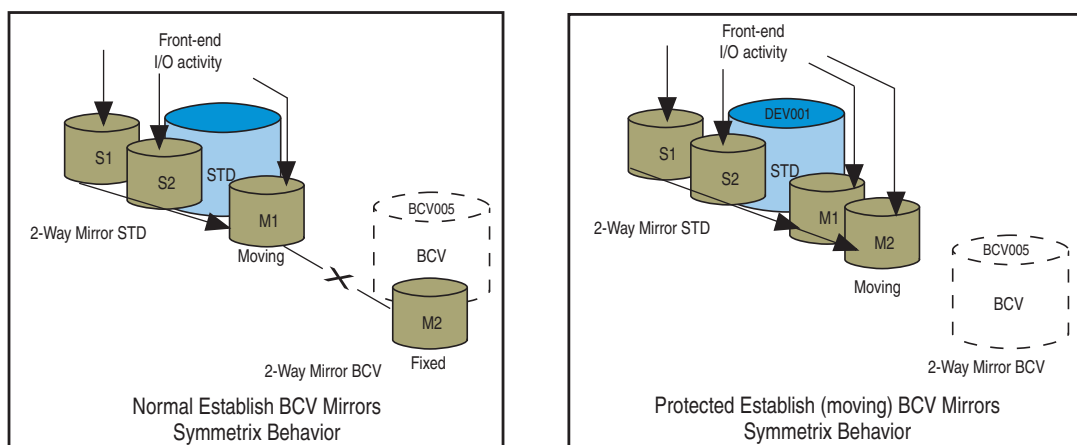


Figure 2-15 Establish Two-Way BCV Mirrors with Protected Establish

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 1d BCV005
```

For more information about the affects of `-protbcvest` during a split operation, refer to *Splitting a Protectively Established BCV* on page 2-60.

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 items 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 2-106).
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 2-107). 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 2-115). 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 2-16 on page 2-41) 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 -g CgName establish
symmir -f[file] FileName establish
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

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 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` or `-opt` options.

Figure 2-16 illustrates the incremental establishing of a BCV pair.

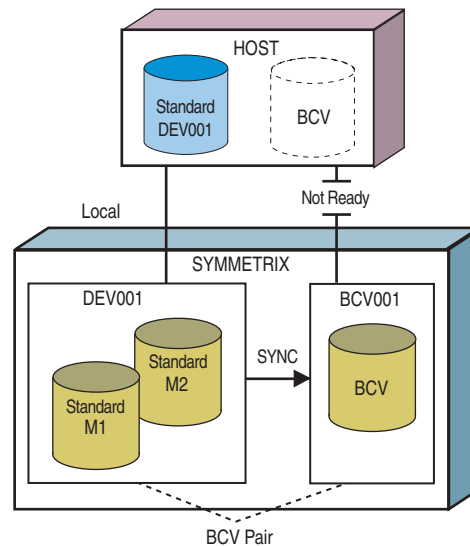


Figure 2-16 Incremental Establish of a BCV Pair

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 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 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.

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.

Establishing Multiple BCV Pairs

To use a BCV device for BC procedures, you must again split the BCV pair to make the BCV device available to its host.

You can incrementally establish (at different times) up to 8 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 (as shown in Figure 2-17), a multi-BCV environment becomes established that retains progressive historical snaps of the data on 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 the 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.

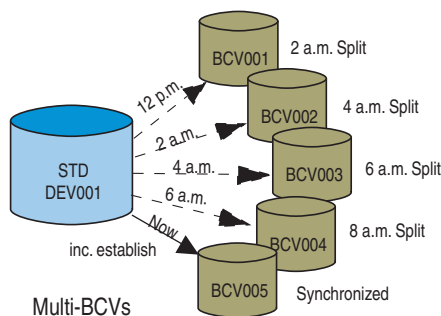


Figure 2-17 Establish a Multi-BCV Environment

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 1d 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 *Restoring (Incremental) Multiple BCV Pairs* on page 2-82).

For instructions on cancelling the incremental relationship to one of the BCVs from the set, refer to *Canceling a Multi-BCV Relationship* on page 2-31.

Incrementally Establishing SRDF-Connected BCV Pairs

You can also 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 2-18 on page 2-44 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.

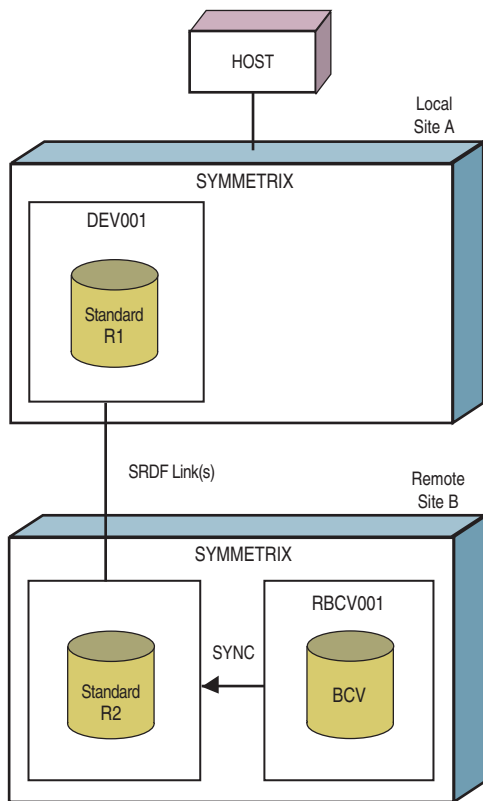


Figure 2-18 Incremental Establish of the Remote BCV to Mirror Local STD

Figure 2-19 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.

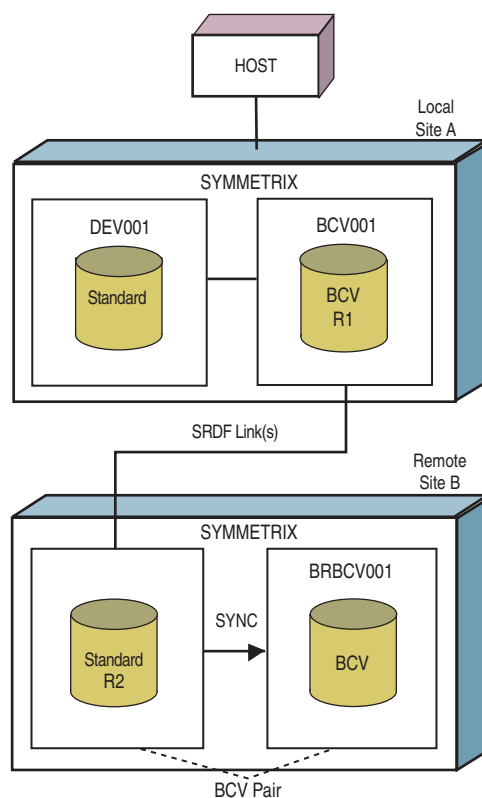


Figure 2-19 Incremental Establish of the Remote BCV to Mirror Local BCV

Note that the base tasks performed with `symbcv` such as `list`, `associate`, and `disassociate` locally or remotely connected BCV devices, are described in the BCV support chapter of the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

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 1d BCV005
```

For more information about the protected BCV established environment, refer to *Protected BCV Establish (Moving All Mirrors)* on page 2-38.

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 -g CgName split
symmir -f[file] FileName split
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

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 2-20 on page 2-48 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 2-127.

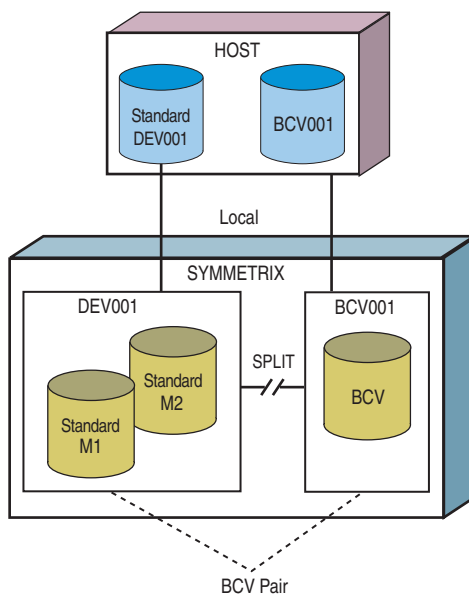


Figure 2-20 Split the BCV Pair

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.
- ◆ I/O is suspended to the standard device until the split operation completes.
- ◆ 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.

Once you finish running any Business Continuity 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 2-21 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.

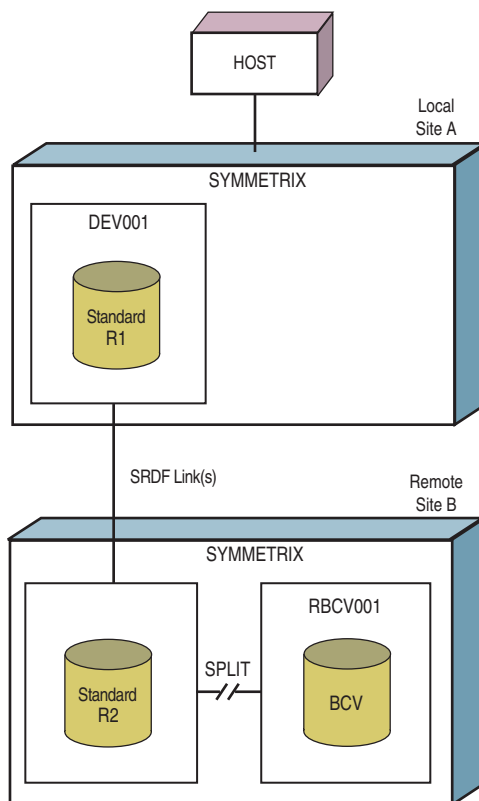


Figure 2-21 Split the Remote BCV Pair That Mirrors the Local STD

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 2-22 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.

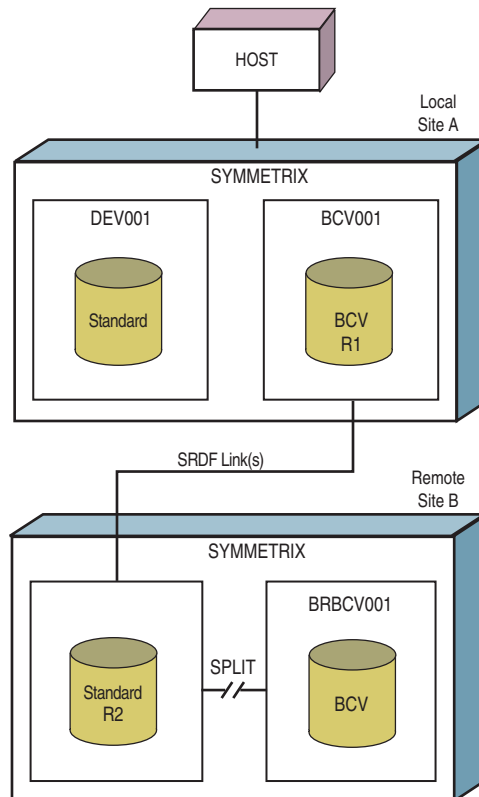


Figure 2-22 Split the Remote BCV Pair That Mirrors the Local BCV

Remote Copy with BCV Split

In addition to splitting a local BCV pair, you can further specify the remote (`-remote`) option which will propagate the R1 BCV copy across the SRDF link to the R2 that mirrors the BCV.

```
symmir -g DgName split DEV001 -remote
```

Figure 2-23 illustrates the remote copy to the R2 mirror from the local BCV split.

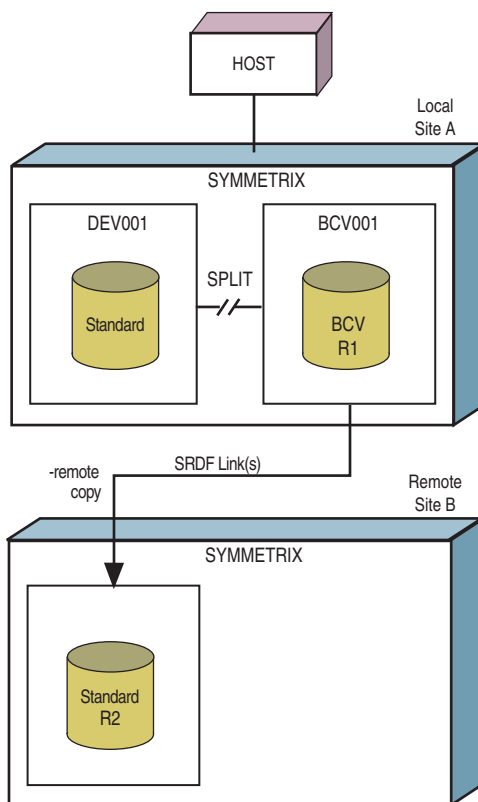


Figure 2-23 Remote Copy with a BCV Split

Differential Split

For Enginuity Version 5x65 only, the differential option (`-diff`) sets the differential split behavior, which synchronizes the primary mirror of a BCV device and its secondary mirror in a differential copy mode. In this way, only tracks that changed on the standard device will propagate after the split to the other mirror of the BCV device, as shown in Figure 2-24.

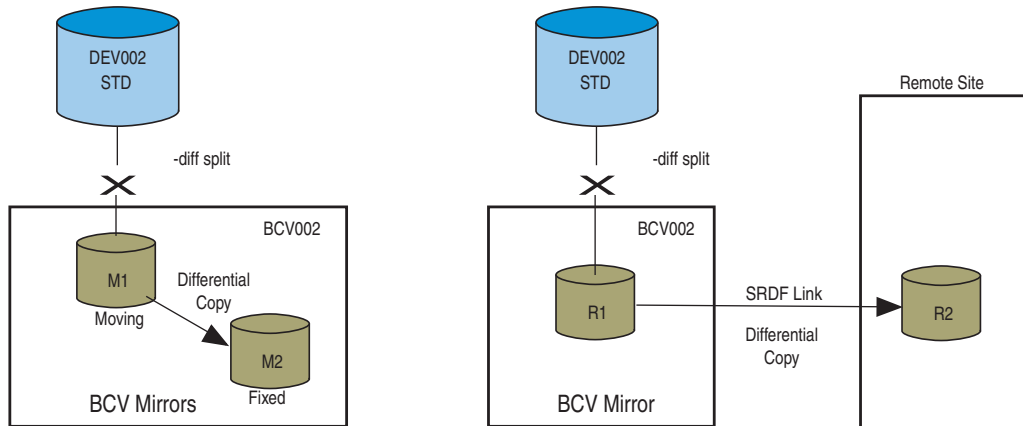


Figure 2-24 Differential Copy in a Differential Split

For example:

```
symmir -g prod -diff split
```

Note: This option is obsolete for Enginuity Version 5x66 and higher as the differential split mode is now the default behavior.

Instant Split

The instant (`-instant`) option improves the performance of a typical split operation by performing a quick foreground BCV split. It returns a more immediate successful status back to the host, allowing applications to continue. The instant split option is supported for Engenuity Versions 5x66 and 5x67. Since Version 5x68, this option is no longer required because instant split mode is the default behavior.

For example, to perform an instant foreground split with device group `Prod`, enter:

```
symmir -g Prod split -instant
```

This option can be made the continual default split mode by setting:

```
SYMAPI_DEFAULT_BCV_SPLIT_TYPE = INSTANT
```

in file:

```
/var/symapi/config/options  
C:\Program Files\EMC\symapi\config\options
```

Note that the instant split option allows you to access the BCV immediately before the split completes in the background.



CAUTION

The following immediate post operations are illegal and can fail, or are improper if the split is not fully complete in the background.

- ◆ You cannot restore any BCV to the standard when the standard has not yet completed the background split.
- ◆ You cannot establish (full or increment) the STD with another BCV device.
- ◆ If you were to incrementally reestablish the splitting pair (while still in background), any tracks that are still splitting in the background will become Invalid on the BCV device and need to be copied.

The wait-for-background-split option (`SYMAPI_WAIT_FOR_BCV_BG_SPLIT`) in the options file can be used when you want your script or application calls to wait for the entire (background) split operation to complete before returning to your script or program.

To make this change so the default behavior of the instant split mode waits for the entire split operation to complete, enter the following in the options file:

```
SYMAPI_WAIT_FOR_BCV_BG_SPLIT = TRUE
```

FALSE is the default.

To remove any of the instant default options, remove the line entry or comment the line out by using a pound sign (#) at the beginning of the line.

To check for a completed instant split (including the background split) before initiating these operations, enter:

```
symmir -g DgName verify -bg -split -i 10
```

For information on the options file, refer to Appendix B in the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

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 5x66 and higher.

Normal Split Behavior

Normally, when a BCV has 2 mirrors, only the primary mirror (M1) joins the standard device in establish or restore operations. As shown in Figure 2-25, 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.

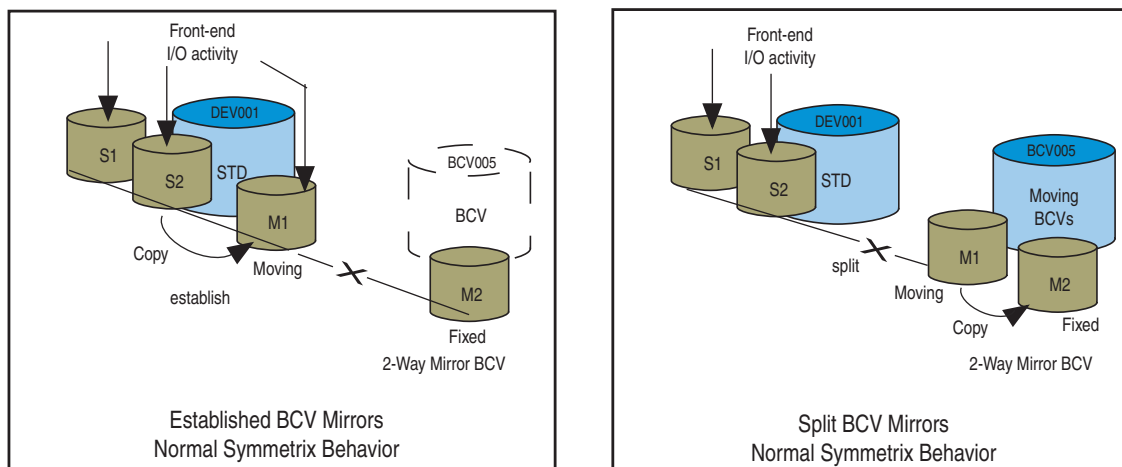


Figure 2-25 Two-Way Mirror BCV Establish/Split Normal Behavior

Reverse Split Behavior

In a reverse split operation, the direction of data flow between the BCV mirrors is reversed. As shown in Figure 2-26 on page 2-57, 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.

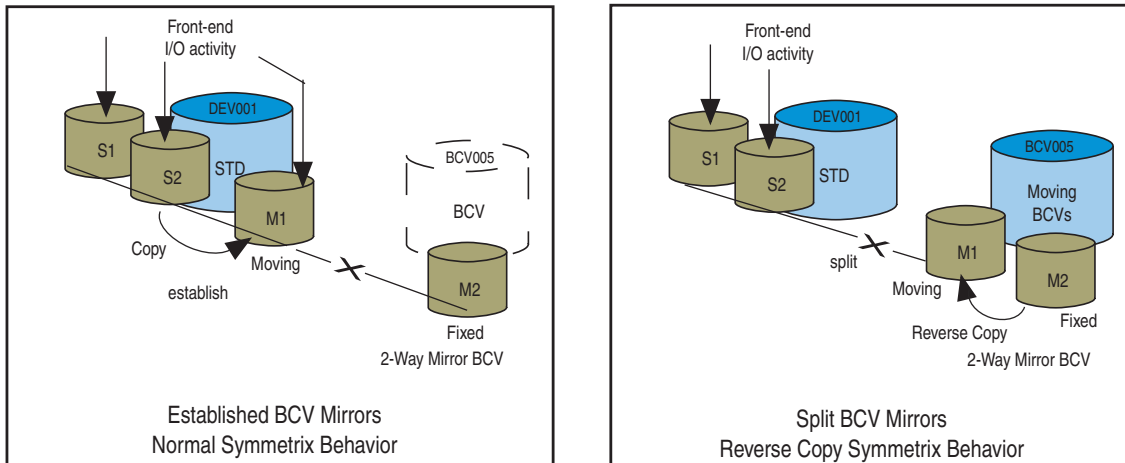


Figure 2-26 2-Way Mirror Reverse Establish/Split Behavior



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:

```
symm1r -g Prod establish -reverse DEV001
symm1r -g Prod split -reverse DEV001
symm1r -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.

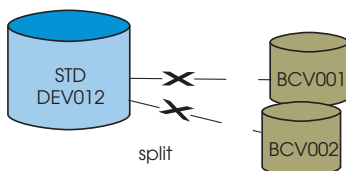
Splitting Concurrent BCV Pairs

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 2-36).

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 (refer to *Instant Split* on page 2-54).

For example, to split concurrently established BCV001 and BCV002 with standard DEV012 in device group CncGrp, enter:

```
symmir -g CncGrp split DEV012
```



Concurrent BCVs

Figure 2-27 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 verify -bg DEV012 BCV 1d 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 2-28 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.

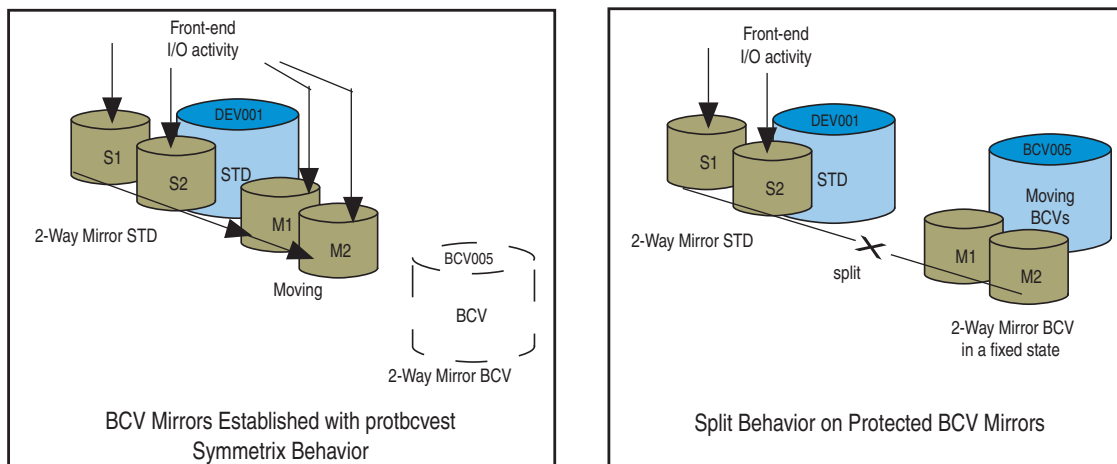


Figure 2-28 Split Behavior on Two-Way BCV Mirrors

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 DgName 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 *Protected BCV Establish (Moving All Mirrors)* on page 2-38.

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
```

Note: For more information on using the `symioctl` command, refer to the *Using SYMCLI to Perform TimeFinder Consistent Splits* white paper.

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 (refer to Figure 2-29 on page 2-63) 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 2-64.

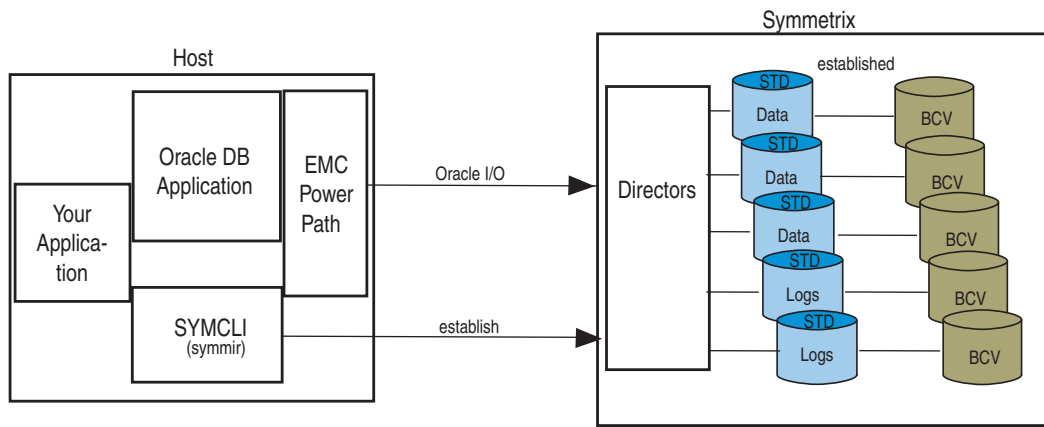


Figure 2-29 An Established Oracle DB Environment

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 or the Enginuity Consistency Assist feature.

This functionality requires a TimeFinder/CG license. For information about consistent split using the Enginuity Consistency Assist feature, refer to *Enginuity Consistency Assist* on page 2-68.

Consistent split operations can also be used in conjunction with SRDF/AR (Automated Replication) to set up automatic remote mirroring according to a predefined copy schedule. For more information, refer to the *EMC Solutions Enabler Symmetrix SRDF CLI Product Guide*.

Consistent Split Using PowerPath

Since Enginuity Version 5x66, 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 V2.0.1 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.

Coding Pre- or Post-Action Scripts

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.

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 2-30 shows the sequences to execute a consistent instant split operation on an Oracle environment.

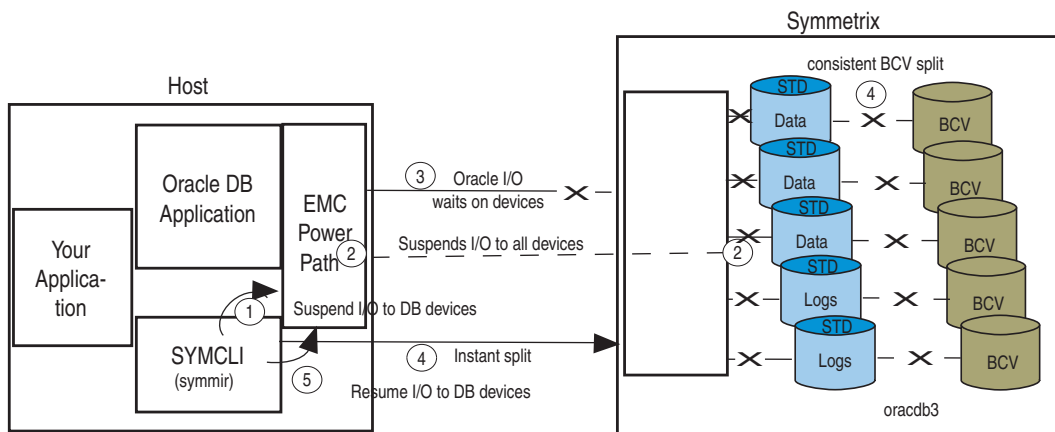


Figure 2-30 Consistency Split on Oracle Environment

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.

TimeFinder Consistent Split on Both RDF Sides

In an RDF environment as shown in Figure 2-31, 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.

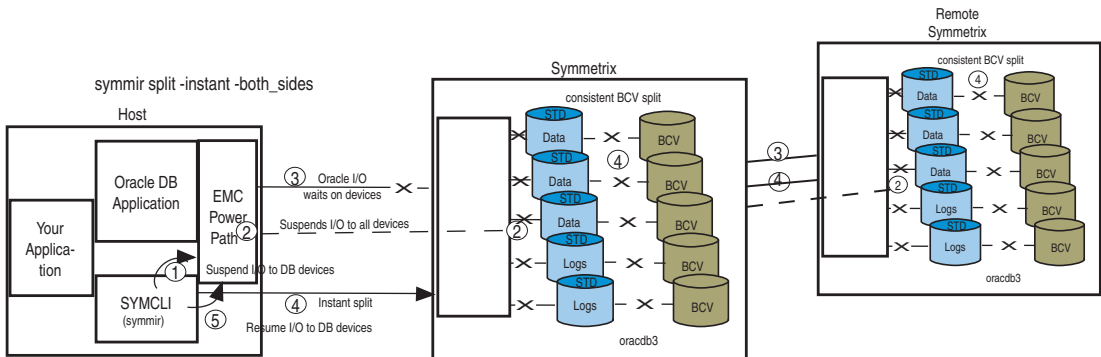


Figure 2-31 Consistent Split on Both RDF Sides

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]
```

Engenuity Consistency Assist

You can use the Engenuity Consistency Assist (ECA) feature 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 2-32 for a depiction of how a control host can perform ECA consistent splits for three database hosts that access devices on a Symmetrix array.

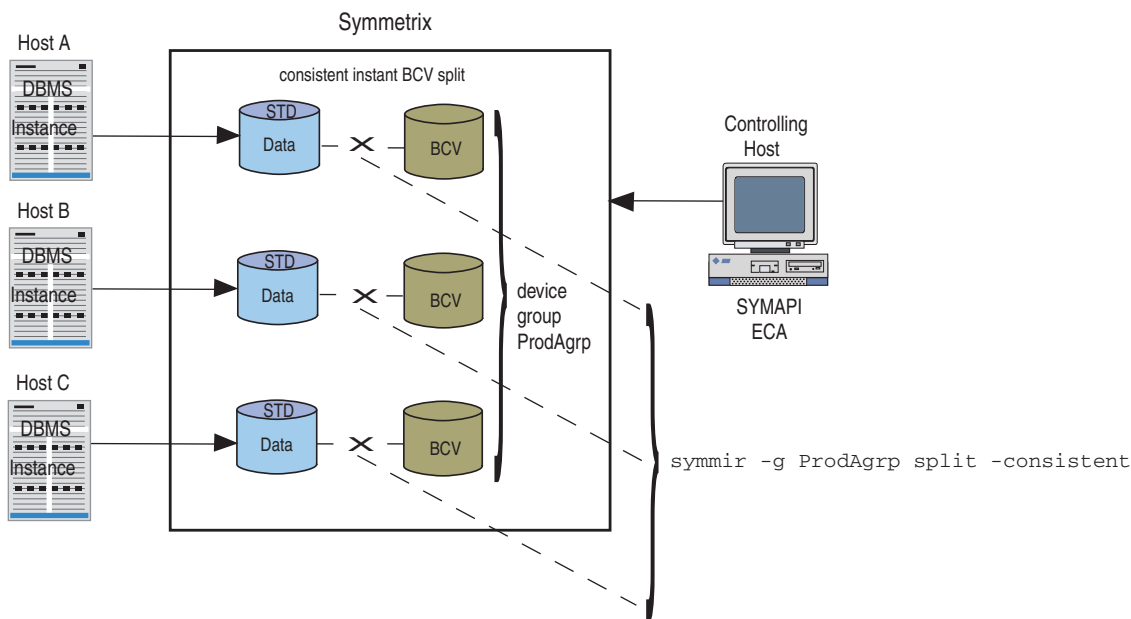


Figure 2-32 ECA Consistent Split

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 2-32), 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.

The following steps explain the example in Figure 2-32 on page 2-68 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.

```
symdmg 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
```

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 30 seconds, 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 2-2 provides a comparison of support criteria characteristics for using Enginuity Consistency Assist (ECA) versus PowerPath.

Table 2-2 Consistent Split Comparison: ECA Versus PowerPath

Support Criteria	-consistent	-ppath
Minimum Enginuity support version	5267	5x66
Minimum Solutions Enabler support version	5.1	4.3.x
Multi-host 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 using ECA allows you to consistently split BCVs from R2 devices operating in asynchronous mode (SRDF/A).

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 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
```

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 -g CgName -full restore
symmir -f[ile] FileName -full restore
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

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 1d BCV001 \
DEV002 BCV 1d BCV002 \
DEV005 BCV 1d BCV003
```

The restoration process (Figure 2-33) is complete when the standard device and BCV device contain identical data.

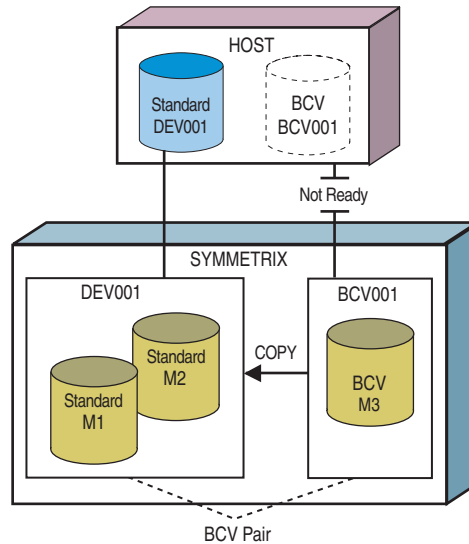


Figure 2-33 Full Restore of the BCV Pair

Note: Except when the **-protect** option is used, 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.

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 2-33, 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 Continuity 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 Continuity 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.

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 2-34 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.

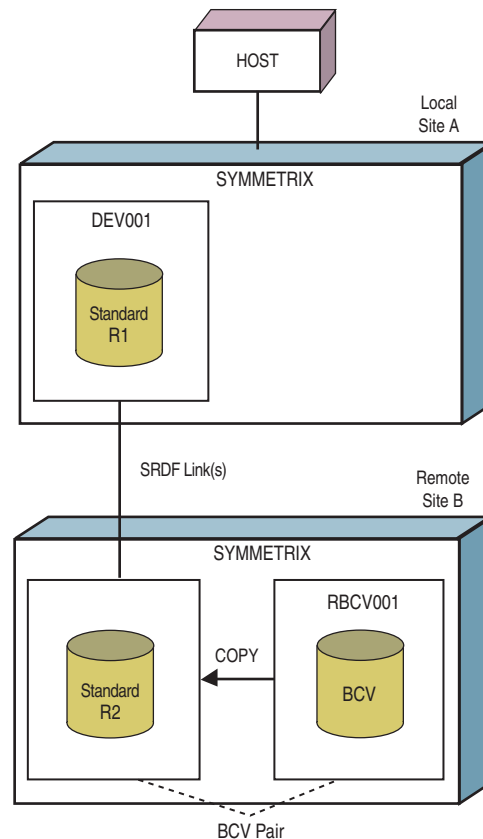


Figure 2-34 Full Restore of Remote STD That Mirrors Local STD

Figure 2-35 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.

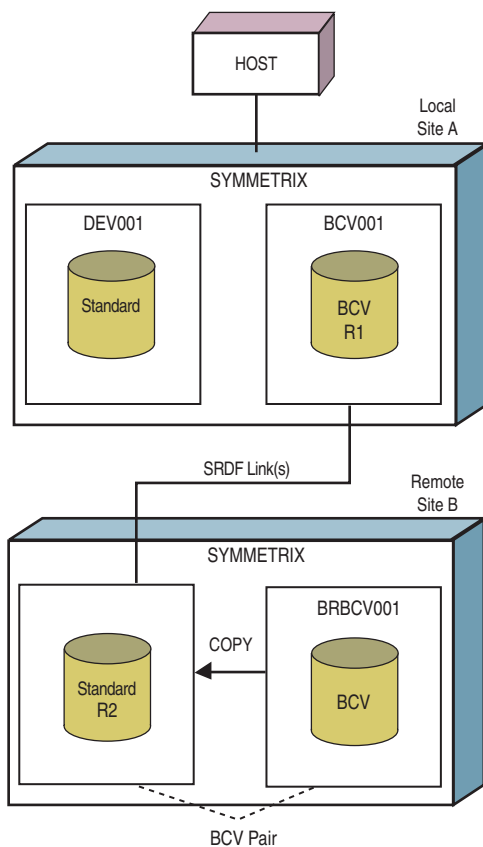


Figure 2-35 Restore of Remote STD That Mirrors Local BCV

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 DgName restore -rdf -bcv -remote
```

Note: Since Enginuity Version 5568, the `-remote` option is longer supported. Use the recommended two step method described in Figure 2-37 on page 2-78.



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.

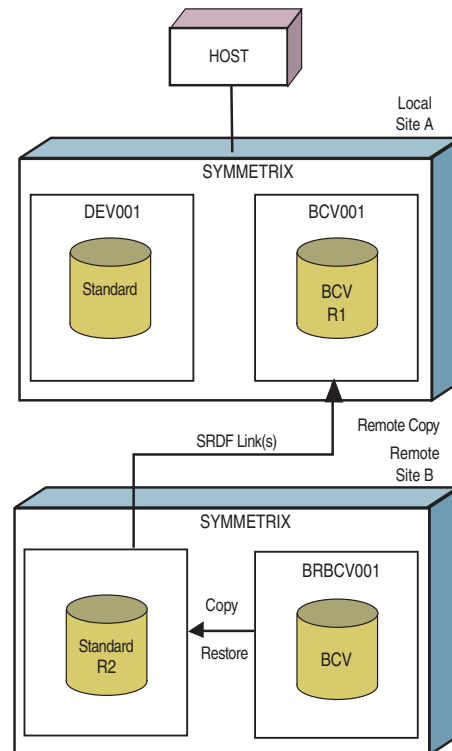


Figure 2-36 Remote Copy with Restore

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
symrdf -g DgName restore
```

This preferred method requires a remote host connection:

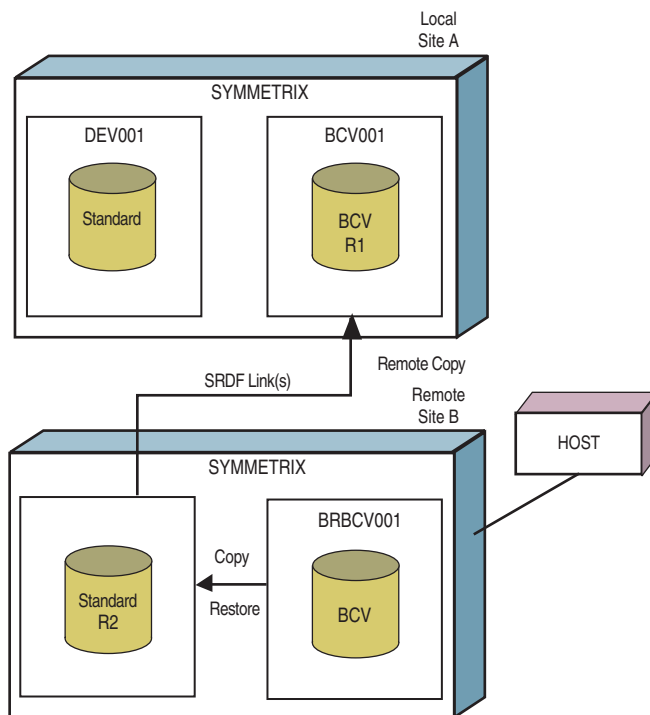


Figure 2-37 Restoring with the Remote Host

Incrementally Restoring BCV Pairs

The incremental restore process (Figure 2-38) 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.

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

```
symmir -g DgName restore
symmir -g CgName restore
symmir -f[file] FileName restore
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

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 2-8.

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 2-38 illustrates an incremental restore of a BCV pair.

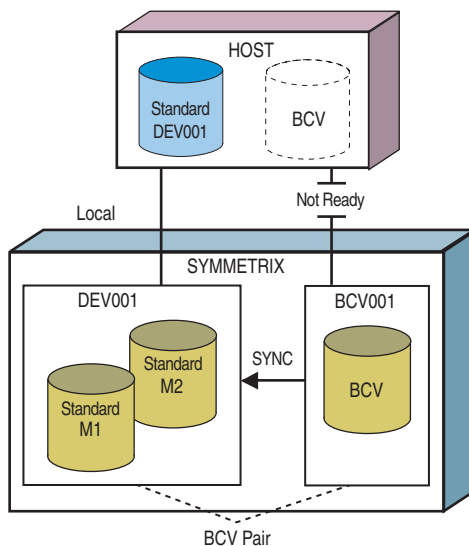


Figure 2-38 Incremental Restore the STD

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.

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.

Restoring (Incremental) Multiple BCV Pairs

You can incrementally establish or restore up to 8 BCV pairs 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 *Establishing Multiple BCV Pairs* on page 2-42), a multi-BCV environment becomes established that retains progressive historical snaps 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 2-39 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.

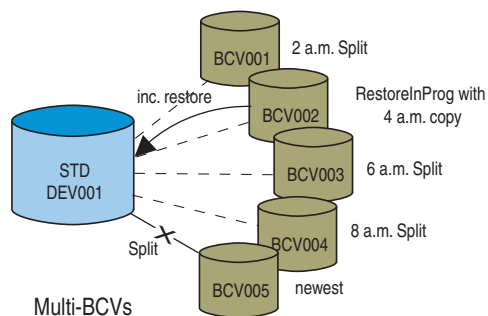


Figure 2-39 Restoring a BCV in a Multi-BCV Environment

For example, to restore the current standard with old data from `BCV002`, enter:

```
symmir -g DgName split DEV001
symmir -g DgName restore DEV001 BCV 1d BCV002
```

Incremental Restore 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.

Note: Since Enginuity Version 5568, the `-remote` option is longer supported. Use the recommended two-step method described in Figure 2-37.



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.

Note: The `-remote` option is not supported in Enginuity Version 5568 and higher.

For example, Figure 2-40 on page 2-84 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.

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
```

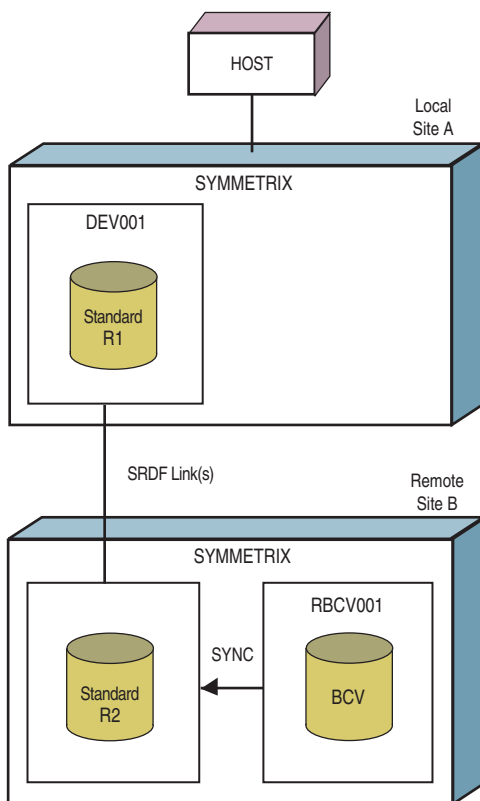


Figure 2-40 Incremental Restore the Remote STD that Mirrors Local STD

Figure 2-41 on page 2-85 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.

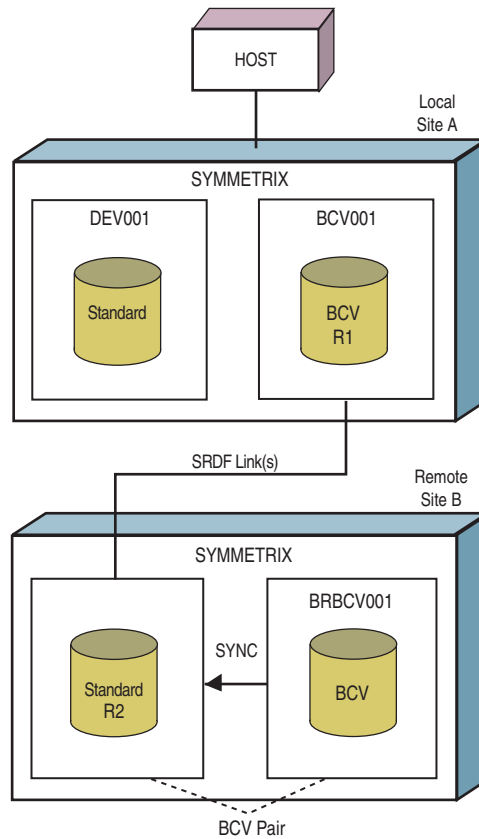


Figure 2-41 Incremental Restore of the Remote STD that Mirrors Local BCV

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.

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 BCV pair. 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 that if you ever need to split again a device 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.

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.

To initiate a preferred attachment on a BCV pair (`DEV001`) in the `prod` group, enter:

```
symmir -g prod attach DEV001 BCV 1d BCV001
```

To initiate a preferred attachment on more than one BCV pair (list) in the `prod` group, enter:

```
symmir -g prod attach DEV001 BCV 1d BCV001 \  
DEV002 BCV 1d BCV002 \  
DEV005 BCV 1d BCV003
```

Note: The `attach` and `detach` preferred relationship are only known to the SYMAPI database on which you are operating.

Figure 2-42 illustrates a preferred attachment of a BCV pair.

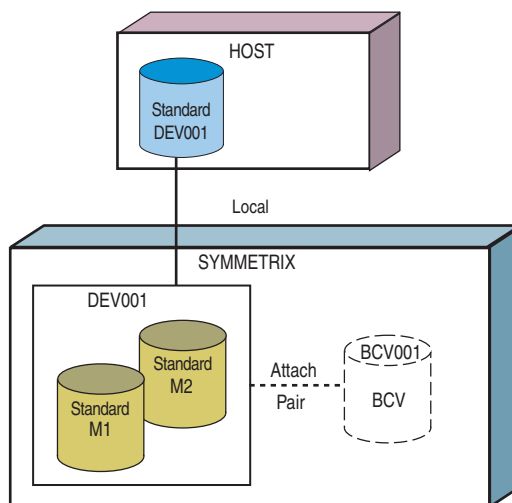


Figure 2-42 Preferred Attachment for BCV Pairing

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, the device names need not be specified.

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 2-43 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 1d 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.

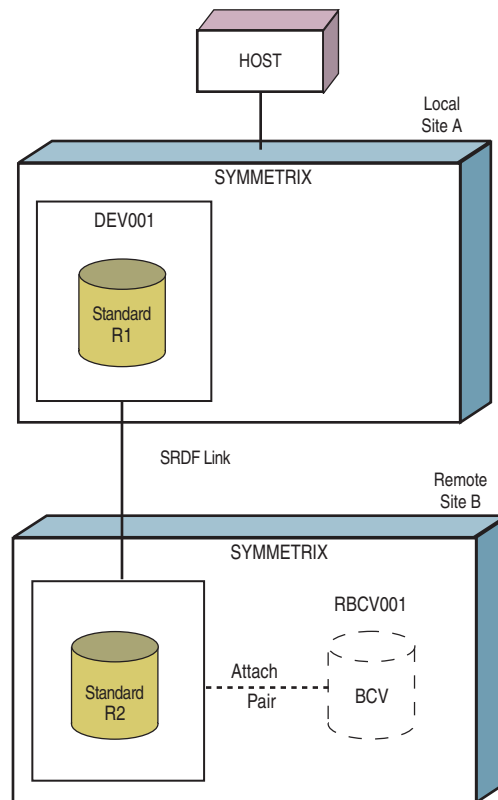


Figure 2-43 Preferred Attachment of Remote BCV to Mirror Local STD

Figure 2-44 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 1d 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.

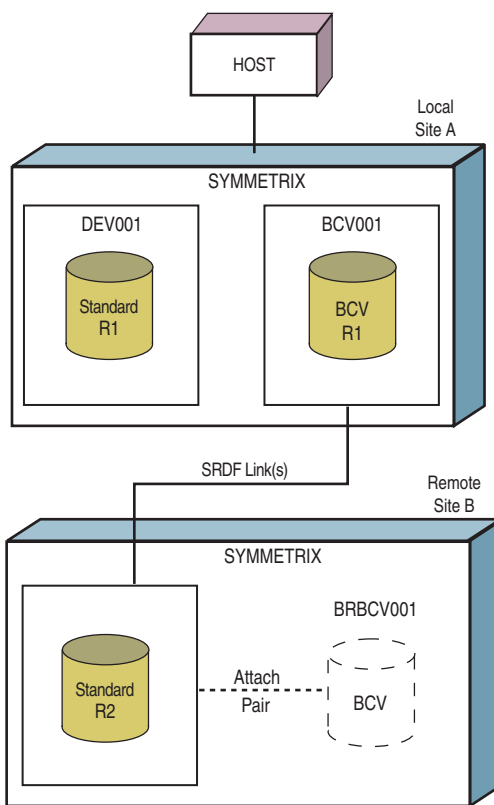


Figure 2-44 Preferred Attachment of Remote BCV to Mirror Local BCV

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.

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 2-45 illustrates a detachment of a BCV pair preference.

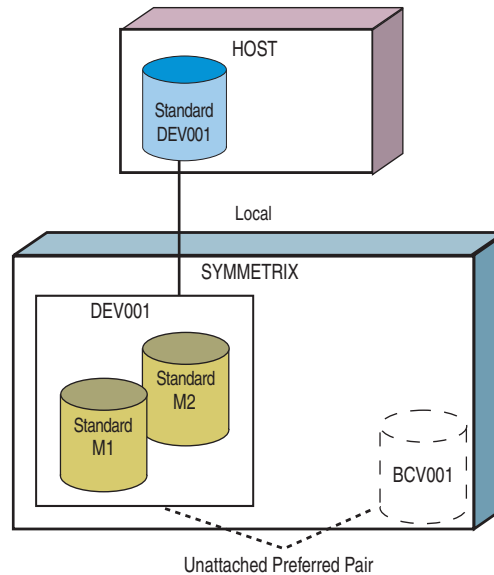


Figure 2-45 Detach a Preferred BCV Attachment

Detaching BCV Preferences for Remote Devices

You can also 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 2-46 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.

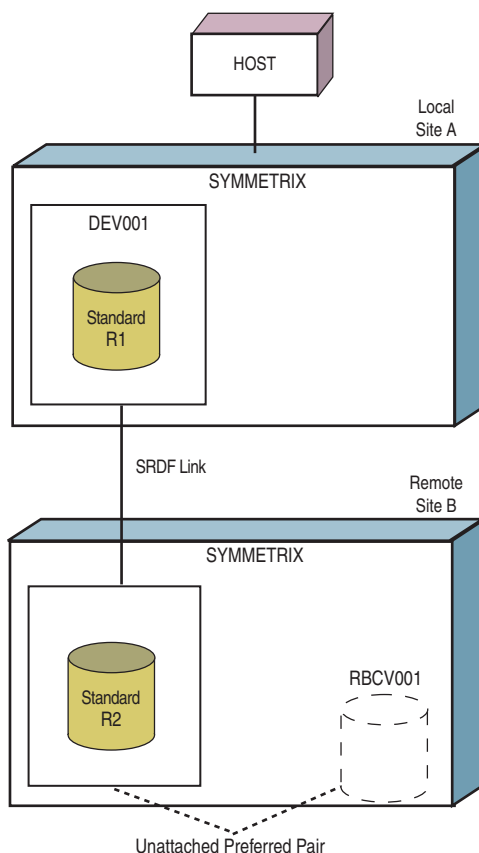


Figure 2-46 Detach a BCV Preference from Remote STD (Mirrored Local STD)

Figure 2-47 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 1d 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.

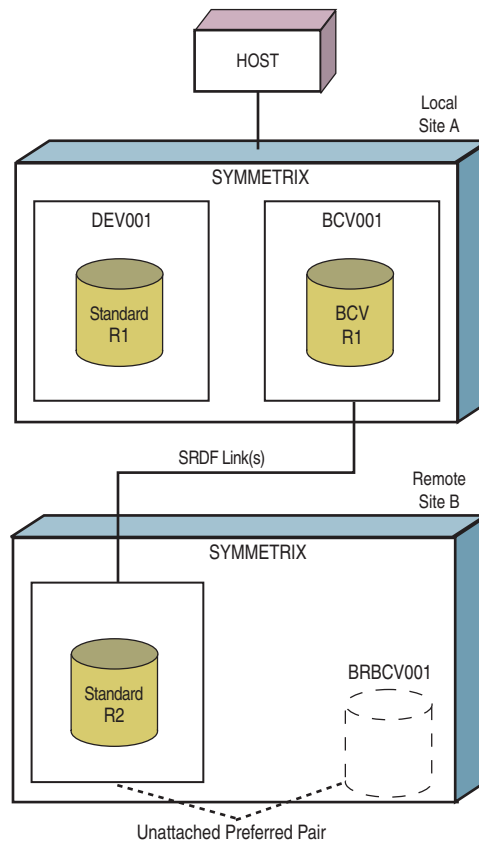


Figure 2-47 Detach a BCV Preference from Remote STD (Mirrored Local BCV)

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:

```
syndg 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 your business applications with incoming database data from certain anticipated peak periods in the day. It is decided that three snaps of your 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 1d
BCV001
```

```
symmir -g MultigrpA split DEV001 #split at 3:10pm
```

```
symmir -g MultigrpA -full establish DEV001 BCV 1d
BCV002
```

```
symmir -g MultigrpA split DEV001 #split at 3:20pm
```

```
symmir -g MultigrpA -full establish DEV001 BCV 1d
BCV003
```

2. It's 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 1d 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 won't be needed.

```
symmir -g MultigrpA cancel DEV001 BCV 1d 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 1d BCV003
```

Querying BCV Pairs

You can perform a query to determine the state of a BCV pair or all BCV pairs in a device group. The query is sent via the gatekeeper device to the Symmetrix array, returning with information about the state of the BCV pair(s).

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

```
symmir -g DgName query
symmir -g CgName query
symmir -f[file] FileName query
```

For more detail about defining a device file, refer to *Device File* on page 2-105.

Query

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 -rdf -rrbcv
```

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 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 and establish operations, enter:

```
symmir -g prod query -bg -percent
```

Verifying BCV Pair States

You can verify whether one or all BCV device pair(s) in a device group are in the Synchronized or Restored states. The `verify` option can be used in scripts to guarantee that the BCV device pair(s) are in the Synchronized or Restored states prior to executing subsequent SYMCLI commands.

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

```
symmir -g DgName verify
symmir -g CgName verify
symmir -f[file] FileName verify
```

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 -rdf -rrbcv verify
```

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.

You can use the `-i` option to repeat the verify actions for a repeat interval (in seconds) so as to be immediately informed when the pairs reach full synchronization. For example:

```
symmir -g prod verify -i 5
```

With this command, the `verify` command keeps repeating every 5 seconds until it finds and reports the pairs fully synchronized.

**Verify Concurrent
BCVs**

For concurrent BCVs, you can verify synchronized or restored states of the two concurrent BCVs. For example:

```
symmir -g prod verify -i 5 -concurrent
```

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` or `symbcv` commands, the BCV state is changed as illustrated in Table 2-3. You will see the abbreviated BCV pair state listed using the SYMCLI commands.

Table 2-3 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	The Invalid BCV pair state only applies to the meta head, where one or more of the devices in a meta BCV pair are in a state other than Synchronized and SyncInProg.

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 2-4).

Table 2-4 Actions for BCV Devices

Argument	Transient State	Final State
<code>establish</code>	SyncInProg	Synchronized
<code>split</code>	SplitInProg	Split
<code>restore</code>	RestInProg	Restored

BCV Actions and Applicable States

Table 2-5 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 2-5 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
<code>establish -full</code>	X				X	X	X ^a	X ^b			X
<code>establish</code>					X						
<code>split</code>		F	X						F	X	
<code>restore -full</code>	X				X	X ^a	X ^b	X ^a			X
<code>restore</code>					X						
<code>attach</code>	X	X	X	X	X	X	X	X	X	X	X
<code>detach</code>	X	X	X	X	X	X	X	X	X	X	X
<code>cancel</code>					X		X	X			

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.

symmir -g Command Line Options

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 2-6 lists the `symmir` control operations and the possible options to use when targeting a specified device group.

Table 2-6 symmir -g Control Arguments and Possible Options

Options	Argument Action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-bcv	X	X	X	X	X	X	X	X
-bypass			X	X	X			
-c, -i	X	X	X	X	X	X	X	X
-concurrent	X	X						
-consistent			X					
-exact	X			X				
-force, -symforce	X	X	X	X	X			
-instant, -both_sides, -diff			X					
-noprompt	X	X	X	X	X	X	X	X
-not_ready			X	X	X			
-opt	X							
-ppath			X					
-preaction, -postaction	X	X	X	X	X			
-preservetgtlocks, -lockid	X	X	X	X	X			
-protbcvest	X	X						
-protect			X	X	X			
-rdb, -dbtype, -db			X					
-rdf	X	X	X	X	X	X	X	X

Table 2-6 **symmir -g Control Arguments and Possible Options (continued)**

Options	Argument Action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-remote			X	X	X			
-reverse	X	X	X	X	X			
-rrbcv	X	X	X	X	X	X	X	X
-skip	X	X	X					
-v	X	X	X	X	X	X	X	X
-vxfs			X					

Table 2-7 lists the `symmir` view arguments and the possible options to use when targeting a specified device group.

Table 2-7 **symmir -g View Arguments and Possible Options**

Options	Argument Action	
	query	verify
-attach	X	
-bcv, -rrbcv	X	X
-bcv_mirrors		X
-bg	X	X
-c, -i	X	X
-concurrent		X
-file	X	X
-force		X
-multi	X	
-offline	X	X
-percent	X	X
-protbcvest	X	
-protect	X	X

Table 2-7 **symmir -g View Arguments and Possible Options (continued)**

Options	Argument Action	
	query	verify
-rdf	X	X
-ready		X
-restinprog		X
-restored		X
-sid	X	X
-split		X
-summary	X	X
-synched		X
-syncinprog		X

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: For more information, refer to *Establishing SRDF-Connected BCV Pairs* on page 2-33.

Both Sides

The both sides (`-both_sides`) option performs an instant consistent split to both the locally and the remotely connected BCV devices (SRDF) within the specified device group.

Bypass The bypass (`-bypass`) option causes the action to bypass host SCSI device reservations.

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
```

Consistency Assist The Engenuity Consistency Assist (`-consistent`) option can be used to perform consistent split operations across multiple, heterogeneous hosts without the use of PowerPath.

Note: For more information, refer to *Engenuity Consistency Assist* on page 2-68.

Databases 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: For more information, refer to *TimeFinder Consistent Split* on page 2-64.

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 multi-hop.

Differential Split For Enginuity Version 5x65 only, the differential (`-diff`) option functions as the differential split flag. It indicates that the split operation should initiate a differential data copy from the moving or primary BCV mirror to the fixed or secondary mirror of the BCV mirror set when the BCV pair split completes.

```
symmir -g prod -diff split
```

Note: Since Enginuity Version 5x66, the differential split behavior is the default, and therefore not required. For more information, refer to *Differential Split* on page 2-53.

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.

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 2-8.

Table 2-8 Results of -force Option

Argument	Result
<code>establish -full</code>	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.
<code>establish</code>	Skip but do not reject devices that are Never Established. Allow devices that are SplitBfrSync.
<code>split</code>	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.
<code>restore -full</code>	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.
<code>restore</code>	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.
<code>verify</code>	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.

Note: For more information, refer to *Instant Split* on page 2-54.

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 standard device in the device group, SYMCLI tries to find 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 are serviced by separate disk adapters.
3. 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.
4. 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.

Note: For more information, refer to *Establishing SRDF-Connected BCV Pairs* on page 2-33.

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: For more information, refer to *TimeFinder Consistent Split* on page 2-64.

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: For more information, refer to *TimeFinder Consistent Split* on page 2-64.

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: For more information, refer to *Protected BCV Establish (Moving All Mirrors)* on page 2-38.

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: For more information, refer to *Protected Restore* on page 2-108.

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.

```
symmir -g prod -rdf 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 a restore operation when the standard device is a remotely mirrored R2 device. If this option is not specified, then the mode defaults to inhibit the remote propagation of data to the remote RDF mirror. For more information, refer to *Using the -remote Option on Multi-Hop Split Actions* on page 2-125.

Table 2-9 shows the impact of the remote option (`-remote`) on the RDF pair states.

Table 2-9 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, it is highly recommended that you perform this action with two commands as follows (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
```

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.

Note: For more information, refer to *Relabeling Devices for Windows 2000* on page 2-127.

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.

```
symmir -g prod -reverse split
```

Note: For more information, refer to *Reverse Split* on page 2-56.

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 SyncInProgress or RestInProgress 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 `SYMCL_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 multi-hop.

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: For more information, refer to *TimeFinder Consistent Split* on page 2-64.

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.

```
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.

```
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.

```
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.

```
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.

For more information, refer to *Instant Split* on page 2-54.

Multi

The multi-BCVs (`-multi`) option directs the `query` operation to the DeltaMark session to display all BCVs that can be incrementally established/restored to the standard device. It lists these devices in chronological order.

symmir -cg Command Line Options

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 2-10 lists the `symmir` control operations and the possible options to use when targeting a specified composite group.

Table 2-10 symmir -cg Control Arguments and Possible Options

Options	Argument Action				
	establish -full	establish	split	restore -full	restore
-bcv	X	X	X	X	X
-bypass			X	X	X
-c, -i	X	X	X	X	X
-concurrent	X	X			
-consistent			X		
-exact	X			X	
-force, -symforce	X	X	X	X	X
-instant, -diff			X		
-noprompt	X	X	X	X	X
-not_ready			X	X	X
-opt	X				
-opt_rag	X				
-ppath			X		
-preaction, -postaction	X	X	X	X	X
-protbcvest	X	X			
-protect			X	X	X
-rdb, -dbtype, -db			X		
-rdf	X	X	X	X	X

Table 2-10 **symmir -cg Control Arguments and Possible Options (continued)**

Options	Argument Action				
	establish -full	establish	split	restore -full	restore
-remote			X	X	X
-reverse	X	X	X	X	X
-rrbcv	X	X	X	X	X
-sid	X	X	X	X	X
-skip	X	X	X		
-v	X	X	X	X	X
-vxfs			X		

Table 2-11 lists the `symmir` view arguments and the possible options to use when targeting a specified composite group.

Table 2-11 **symmir -cg View Arguments and Possible Options**

Options	Argument Action	
	query	verify
-attach	X	
-bcv, -rrbcv	X	X
-bcv_mirrors		X
-bg	X	X
-c, -i	X	X
-concurrent		X
-force		X
-multi	X	
-offline	X	X
-percent	X	X
-protect	X	
-rdf	X	X

Table 2-11 **symmir -cg View Arguments and Possible Options (continued)**

Options	Argument Action	
	query	verify
-restored		X
-sid	X	X
-split		X
-synched		X
-syncinprog		X
-ready		X
-restinprog		X
-protbcvest	X	
-summary	X	

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 2-107).

For all other remaining option descriptions, refer to *Control Options* on page 2-103.

symmir -file Command Line Options

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 multi-hop SRDF link.

For more detail about defining a device file, refer to *Device File* on page 2-105.

Table 2-12 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 2-12 symmir -file Control Arguments and Possible Options

Options	Argument Action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
-bypass			X	X	X			
-c, -i	X	X	X	X	X	X	X	X
-consistent			X					
-force, -symforce	X	X	X	X	X			
-instant, -diff			X					
-noprompt	X	X	X	X	X	X	X	X
-not_ready			X	X	X			
-ppath			X					
-preaction, -postaction	X	X	X	X	X			
-preservetgtlocks -lockid	X	X	X	X	X			
-protbcvest	X	X						
-protect				X	X			
-rdb, -dbtype, -db			X					

Table 2-12 `symmir -file` Control Arguments and Possible Options (continued)

Options	Argument Action							
	establish -full	establish	split	restore -full	restore	attach	detach	cancel
<code>-reverse</code>	X	X	X	X	X			
<code>-skip</code>	X	X	X					
<code>-v</code>	X	X	X	X	X	X	X	X
<code>-vxfs</code>			X					

Table 2-13 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 2-13 `symmir -file` View Arguments and Possible Options

Options	Argument Action	
	query	verify
<code>-attach</code>	X	
<code>-bcv_mirrors</code>		X
<code>-bg</code>	X	X
<code>-c, -i</code>	X	X
<code>-concurrent</code>		X
<code>-force</code>	X	X
<code>-multi</code>	X	
<code>-offline</code>	X	X
<code>-percent</code>	X	X
<code>-protbcvest</code>	X	
<code>-protect</code>	X	
<code>-rdf</code>	X	X
<code>-ready</code>		X
<code>-restinprog</code>		X

Table 2-13 **symmir -file View Arguments and Possible Options (continued)**

Options	Argument Action	
	query	verify
-restored		X
-split		X
-summary	X	
-synched		X
-syncinprog		X

Note: The Symmetrix ID option (-sid) is required for all `symmir -file` commands.

Various Remote Multi-Hop Configurations

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

As shown in Figure 2-48 on page 2-121, you can have multiple sites (for example, Remote Sites B and C) on SRDF links to remotely mirror a local Symmetrix array at Site A. Remote Site B, functioning as a remote mirror to the standard devices at Site A, is most typical. You then can have a third site on an SRDF link (remote Site C) to remotely mirror just the BCV devices in the Symmetrix array at Site A.

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

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
```

At this point, all these devices must be established with the `symmir` and `symrdf` commands. The following set of examples illustrate how various `symmir` and `symrdf` commands might be applied to split operations throughout a complex remote configuration, such as the one shown in Figure 2-48 on page 2-121.

System-Wide Splits

For example, to split the BCV pairs within Site A, enter:

```
symmir -g prod split
```

To split SRDF pairs at Site B from host-connected Site A, enter:

```
symrdf -g prod split
```

To split the BCV pairs within Site B, enter:

```
symmir -g prod -rdf split
```

To split BCV SRDF pairs at Site C from host-connected Site A, enter:

```
symrdf -g prod -bcv split
```

To split the BCV pairs within Site C, enter:

```
symmir -g prod -rdf -bcv split
```

To split BCV SRDF pairs at Site D from Site B, enter:

```
symrdf -g prod -rbcv split
```

To split the BCV pairs within Site D, enter:

```
symmir -file dfile -sid 0014 split
```

or

```
symmir -rrbcv split
```

To split BCV SRDF pairs at Site E from Site C, enter:

```
symrdf -g prod -brbcv split
```

To split the BCV pairs within Site E, enter:

```
symmir -file dfile -sid 0015 split
```

Other operations for these remote sites such as establish and restore apply and execute in the same manner.

For more detail information about `symrdf`, refer to the *EMC Solutions Enabler Symmetrix SRDF CLI Product Guide*.

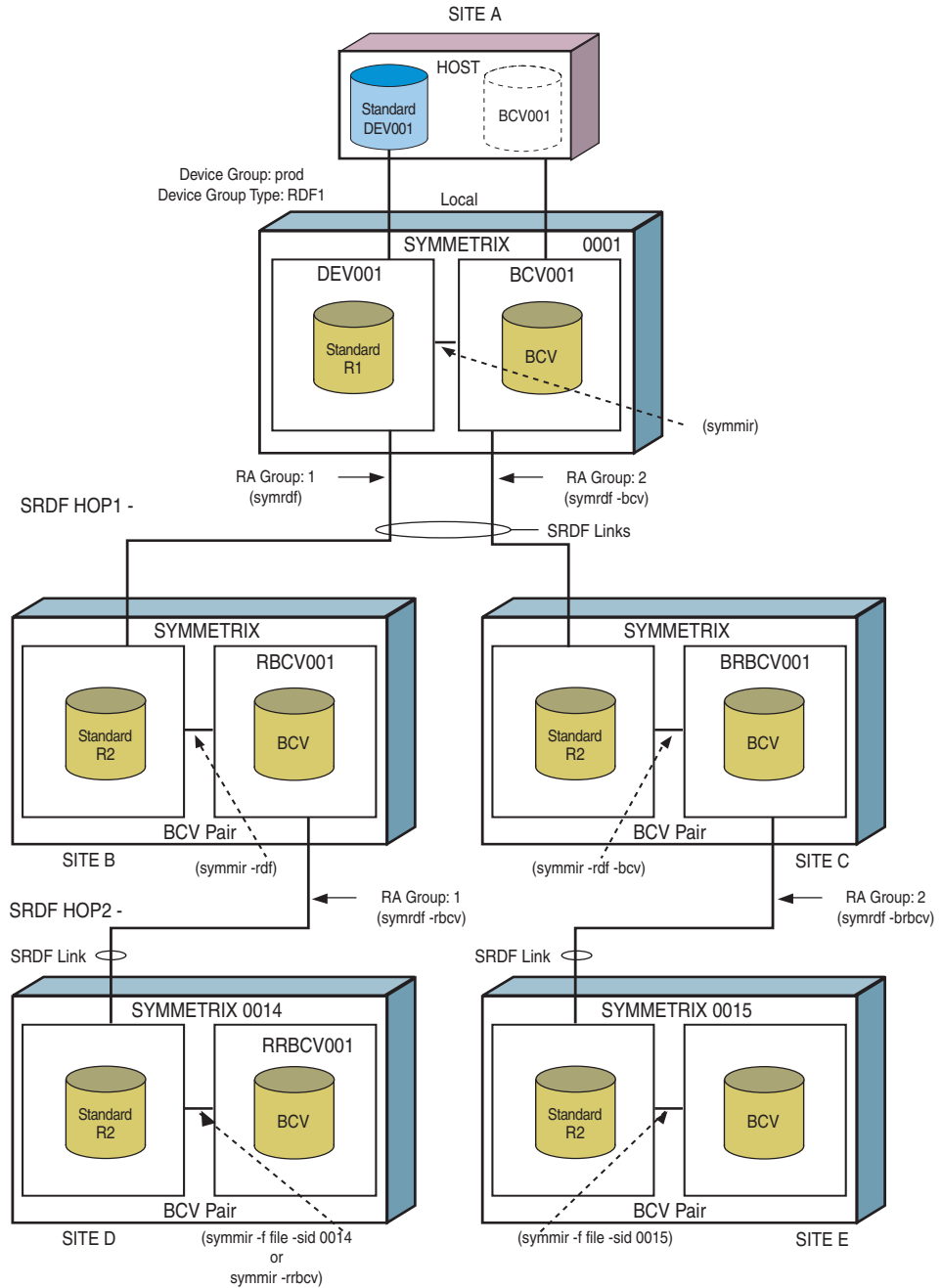


Figure 2-48 Various Remote Configurations

Commands to Various Multi-Hop Devices and Links

This section describes the command application of targeting the various devices and links in complex multi-hop SRDF environments.

The following sequence of commands steps through some basic control operations that touch every device and RDF link in a complex multi-hop configuration. This numbered list works with and is illustrated by Figure 2-49 on page 2-123. The following numbering of commands directly associates with the bubble numbers shown in the figure.

- | | | |
|----|---|---|
| 1 | <code>symrdf -g <> establish</code> | Creates the standard-associated hop 1 copy. |
| 2 | <code>symmir -g <> split -rdf</code> | Splits the standard-associated hop 1 BCV device pair. |
| 3 | <code>symrdf -g <> establish -rbcv</code> | Creates the standard-associated hop 2 copy. |
| 4 | <code>symrdf -g <> restore -rbcv</code> | Restores the standard-associated hop 1 BCV with the hop 2 copy. |
| 5 | <code>symmir -g <> restore -rdf</code> | Restores the standard-associated hop 1 copy with the hop 1 BCV. |
| 6 | <code>symrdf -g <> restore</code> | Restores the standard device with the hop 1 copy. |
| 7 | <code>symmir -g <> split</code> | Splits the standard/BCV pair. |
| 8 | <code>symrdf -g <> establish -bcv</code> | Creates the BCV-associated hop 1 remote copy. |
| 9 | <code>symmir -g <> split -rdf -bcv</code> | Splits the BCV-associated hop 1 device pair. |
| 10 | <code>symrdf -g <> establish -brbcv</code> | Creates the BCV-associated hop 2 copy. |
| 11 | <code>symrdf -g <> restore -brbcv</code> | Restores the BCV-associated hop 1 BCV with the hop 2 copy. |
| 12 | <code>symmir -g <> restore -rdf -bcv</code> | Restores the BCV-associated hop 1 copy with the hop 1 BCV. |
| 13 | <code>symrdf -g <> restore -bcv</code> | Restores the BCV device with the hop 1 copy. |
| 14 | <code>symmir -g <> restore</code> | Restores the standard device with the BCV copy. |
| 15 | <code>symmir -file <> -sid 056 establish</code>
or
<code>symmir -g <> -rrbcv establish</code> | Creates the BCV-associated hop 2 BCV copy. |
| 16 | <code>symmir -file <> -sid 056 split</code>
or
<code>symmir -g <> -rrbcv split</code> | Splits the BCV-associated hop 2 device pair. |

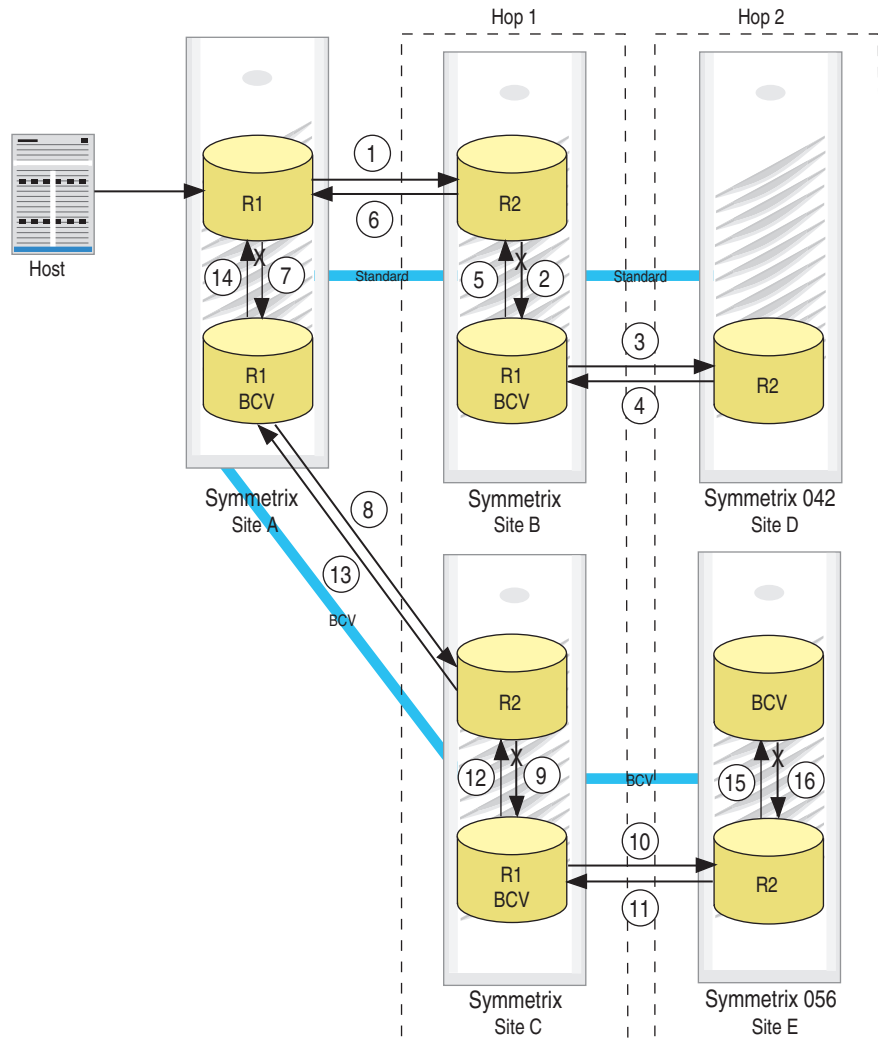


Figure 2-49 Control Operations on Multi-Hop SRDF Configurations

Second-Level Controls for Multi-Hop SRDF Environments

As previously described, second-level multi-hop control operations were accomplished using the device file (`-file`) option for managing RDF BCV device pairs. Since Solutions Enabler V5.3, you can use the `symmir` command for device groups (`-g`) for BCV control capability in second-level multi-hop SRDF environments.

The remote RDF BCV (RRBCV) devices must have been previously associated with the device group using the `symbcv -rrdf` command. For specific information about how to associate second-level multi-hop BCVs with a device group, refer to *Associating Devices in a Compounded Remote Configuration* on page 2-18.

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 multi-hop BCV control operations available with the `symmir` command include `query`, `verify`, `attach`, `detach`, and `cancel`.

Note: Since Solutions Enabler V5.4, the `symmir` command and the `-rrbcv` option have been extended to work for composite groups (`-cg`).

For an example of a BCV establish operation in a second-level multi-hop SRDF environment, refer to *Establish Second-Level Remote BCV Pairs* on page 2-35.

Using the -remote Option on Multi-Hop Split Actions

This section describes the command application of targeting the various devices and links with the `-remote` option in complex multi-hop SRDF environments.

The following sequence of commands steps through some basic control operations that touch every device and RDF link in a complex multi-hop configuration. This numbered list works with and is illustrated by Figure 2-50. The following numbering of commands directly associates with the bubble numbers shown in the figure.

- | | | |
|---|---|---|
| 1 | <code>symrdf -g <> establish</code> | Creates the standard-associated hop 1 copy. |
| 2 | <code>symmir -g <> split -rdf -remote</code> | Splits the standard-associated hop 1 BCV device pair and creates a standard-associated hop 2 copy of the hop 1 BCV. |
| 3 | <code>symmir -g <> split -remote</code> | Splits the standard/BCV pair and creates a BCV-associated hop 1 copy of the local BCV. |
| 4 | <code>symmir -g <> split -rdf -bcv -remote</code> | Splits the BCV-associated hop 1 BCV device pair and creates a BCV-associated hop 2 copy of the hop 1 BCV. |
| 5 | <code>symmir -g <> split -rdf -rrbcv</code> | Splits the BCV-associated hop 2 BCV device pair. You cannot use the <code>-remote</code> option here. |

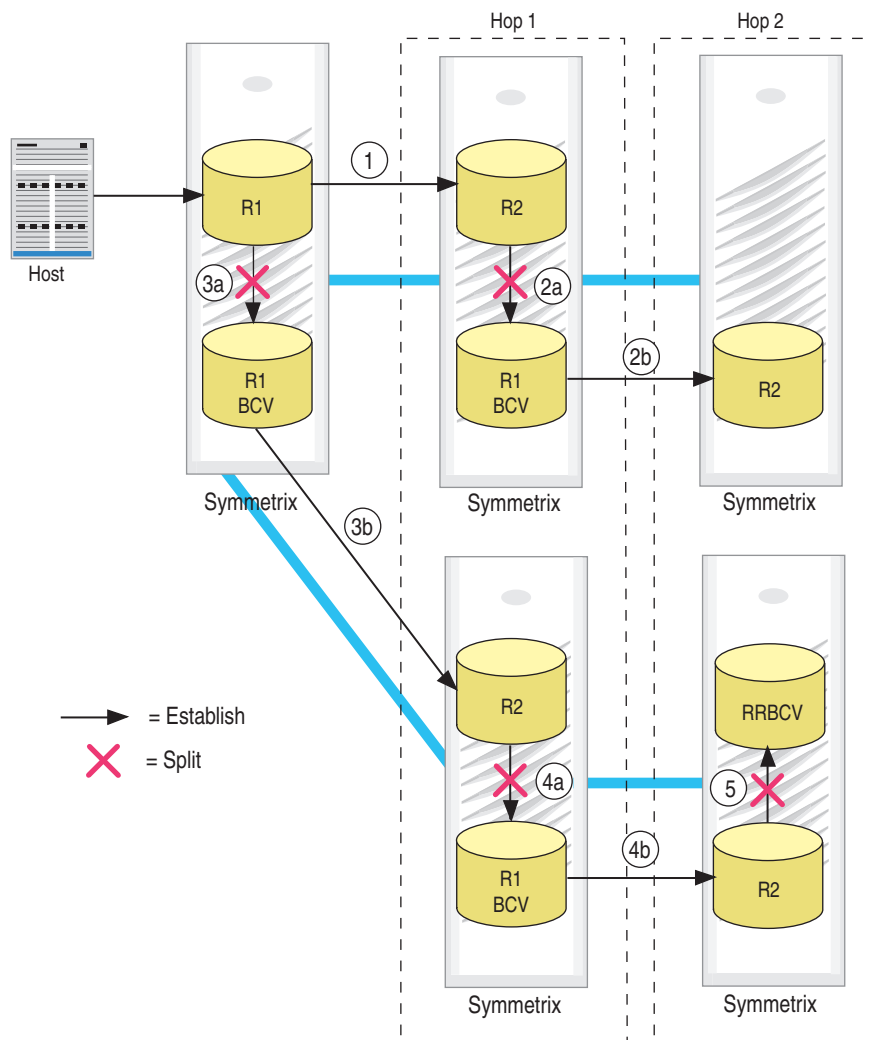


Figure 2-50 The -remote Option on Multi-Hop Configurations

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`

Reference documentation for the first four commands can be found in the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

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 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
```

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

◆ TimeFinder/Clone Environment	3-2
◆ Creating a Clone Copy Session	3-5
◆ Modifying a Clone Copy Session	3-9
◆ Activating a Clone Copy Session	3-10
◆ Recreating a Clone Copy Session	3-12
◆ Restoring Data from a Target Device	3-13
◆ Terminating a Clone Copy Session	3-14
◆ BCV as the Clone Source	3-15
◆ Creating Multiple Clone Copies	3-18
◆ <code>symclone -g</code> Command Line Options	3-23
◆ <code>symclone -file</code> Command Line Options	3-25
◆ <code>symclone -cg</code> Command Line Options	3-27

TimeFinder/Clone Environment

Since Enginuity Version 5568, Symmetrix TimeFinder/Clone operations using the SYMCLI allow you to create *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 meta devices can also be created, but the source and target meta devices 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.

Note: Since Enginuity Version 5669, you can also perform TimeFinder/Snap operations that create virtual Snap copies using the `symsnap` command. For information on this functionality, refer to Chapter 4.

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 may have as many as 16 Clone copies. When using the `-copy` option, you can copy up to four full data copies simultaneously, without disruption to database production activity. For more information on the number of available copies, refer to *Copy Session Limits* on page 3-5.

Clone Operations Overview

TimeFinder/Clone functionality is controlled via 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*, and *terminate* the copy sessions. This chapter explains how to manage the devices participating in a copy session using SYMCLI.

Figure 3-1 illustrates a copy session where the controlling host creates a Clone copy of standard device DEV001 on target device DEV005.

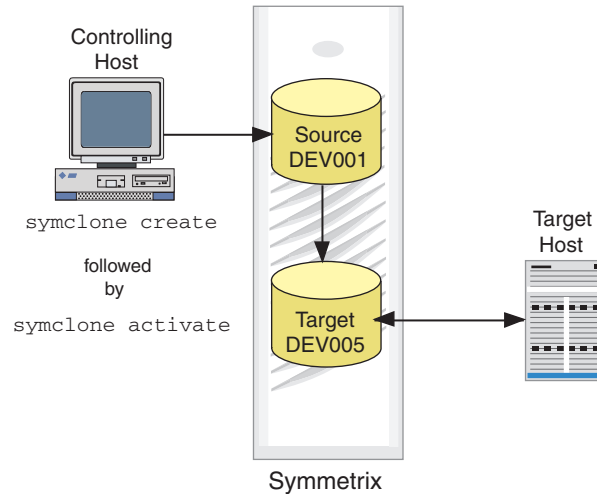


Figure 3-1 TimeFinder/Clone Copy of a Standard Device

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. Listed here are rules that 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.
- ◆ A TimeFinder standard device cannot be created or activated in a copy session if the BCV pair state is `SplitBfrRest` or `RestInProg`.
- ◆ The `symclone terminate` command is allowed for all TimeFinder BCV pair states.

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

For a description of each BCV pair state, refer to Table 2-3 on page 2-99.

For more specific information regarding possible SRDF pair state conflicts, refer to *State Rules for TimeFinder/Clone Operations* on page A-4.

Creating a Clone Copy Session

Initially, you must create a Clone session that defines and sets up the cloning devices you have selected for the Clone operation. For example, to begin a 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 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 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. For more information, refer to *Activating a Clone Copy Session* on page 3-10.

Note: If a copy session is created and not activated, it may be terminated. Although, the data on the target device should then be considered invalid.

Copy Session Limits

The Symmetrix array is currently limited to 16 sessions per source device, which can be used for TimeFinder Mirror, Clone, Snap, or SDDF operations. This limits the number of available Clone copies that can be created. A total of 16 concurrent CopyOnAccess copy sessions can be created from a standard device (sessions created without using the `-copy` option).

However, this number is decremented for each current TimeFinder Mirror, Clone, Snap, and SDDF session. For example, if you have a standard device with two paired BCV devices, you can only create up to 14 copy sessions for that standard device.

Copy sessions created using the `-copy` option to create full data copies are limited to four concurrent sessions, with up to an additional 12 CopyOnAccess sessions, for a total of 16 sessions.

CopyOnAccess

By default, when a copy session is finally activated, the device pair state will be 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.

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.

Full Device 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.

By omitting the `-copy` option (default), the device pair state will be in the 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. For more information, refer to *CopyOnAccess* earlier on this page.

Note: The full copy option (`-copy`) is only applicable when the target is a regular device and not a virtual device.

Precopy Option

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

When using this option, a point-in-time copy cannot be established because the session is not 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 takes over and the pre-copy operation ceases.

Differential Option

The `-differential` option is used with either the `-copy` or `-precopy` option to create an SDDF session for maintaining changed track information. Using the `-differential` option without the `-copy` or `-precopy` option implies a `-copy`. The `-differential` option must be used when creating copy sessions on which you plan on issuing `recreate` or `restore` arguments. For more information on the `recreate` argument, refer to *Recreating a Clone Copy Session* on page 3-12. For more information on `restore` argument, refer to *Restoring Data from a Target Device* on page 3-13.

Copy Functionality by Target Type

Table 3-1 compares copy functionality between TimeFinder/Clone operations (`symclone`) for standard and BCV devices with TimeFinder/Snap operations (`symsnap`) for virtual devices.

Table 3-1 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	No
Copy on first write to source	Yes	Yes
Copy on first write to target	Yes	Yes
Use of the <code>-copy</code> option	Yes	No
Incremental restore	No	Yes
Full restore	No	Yes

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.

Establish Command

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

For example, to create and then activate the copy session shown in the example on page 3-6, enter:

```
symclone -g ProdDB establish DEV001 sym 1d DEV005 -full
```

Note: This implies the `-copy` and `-diff` options.

Modifying a Clone Copy Session

You can use the `set mode` argument to modify the mode in which a copy session is operating after creation or recreation. However, you must use this argument before activating the session.

Possible values are:

- ◆ **Copy** — If the 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.
- ◆ **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 CopyOnAccess once the session is activated and no full device copy will initiate.
- ◆ **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 cannot change to No-Copy mode. Once the session is activated, the session changes to Copy mode. In other words, setting the mode to `precopy` implies a `copy`. For more information on Precopy, refer to *Precopy Option* on page 3-6.

The `set mode` argument can only be used on a clone pair whose state is Created or Recreated.

For example, to modify the copy session created in the example on page 3-6 from Copy mode to No-Copy mode, enter:

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

Activating a Clone Copy Session

To activate the copy session created in the example on page 3-6, 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.

Not Ready Option

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 the `symld ready` command.

Consistent Option

The `symclone activate` command can be used with the `-consistent` option to invoke the Enginuity Consistency Assist (ECA) feature. 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 ECA can be used for consistent split operations, refer to *Enginuity Consistency Assist* on page 2-68.

Note: Since Enginuity Version 5x67, 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. Refer to Figure 3-2, 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.

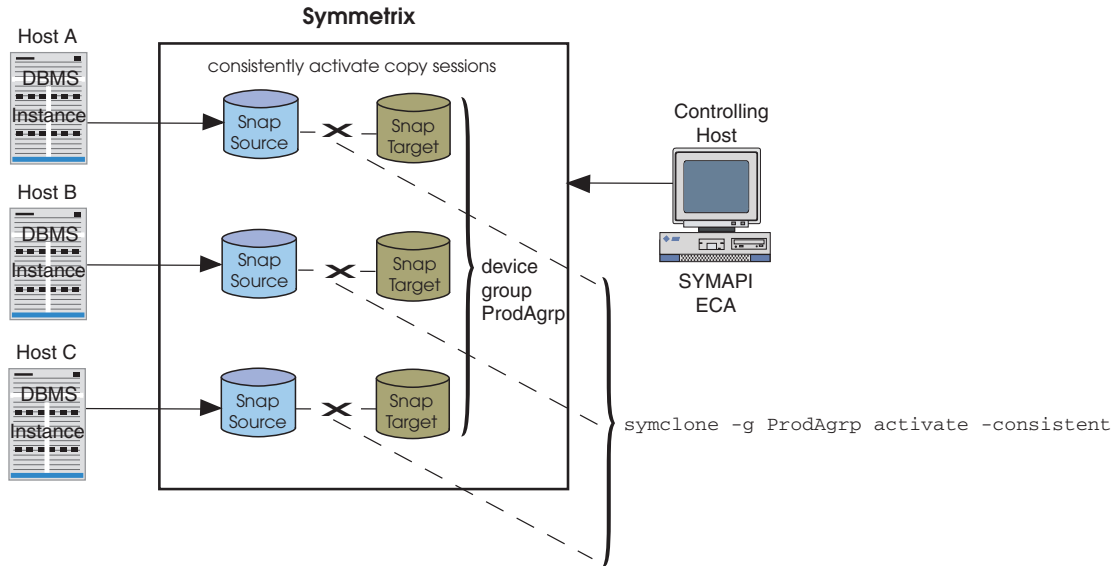


Figure 3-2 TimeFinder/Clone Consistent Activate Using ECA

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 3-2), 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.

Recreating a Clone Copy Session

Once a clone session is fully copied, the `symclone recreate` command allows you 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 Clone 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 and to initiate the copy of the changed data.

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

This feature uses SDDF to keep track of the changed data, which requires a new SDDF option to be open on the source device. Therefore, the maximum number of clone sessions is decreased from 16 to 15.

For example, to recreate the clone session created in the example on page 3-6, enter:

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

Precopy Option

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

When using this option, a point-in-time copy cannot be established because the session is not 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.

Establish Command

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

For example, to recreate and then activate the copy session shown in the example on page 3-6, enter:

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

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 will terminate and an incremental copy session back to the original source device will start. 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 target device created in the example on page 3-6 to device `DEV006`, enter:

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

Terminating a Clone Copy Session

To terminate the copy session activated in the previous example 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 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 pair states.

Note: A created and activated copy session may be terminated, but the data on the target device will become invalid (whether or not data was actually copied to the target).

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.

BCV as the Clone Source

As shown in Figure 3-3, 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 2.

Note: For this example, with other hosts having access to the BCV source, it might be wise to make the BCV source Not Ready when you perform the initial split.

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: Upon completion of the copy session, you may need to release any Not Ready state you imposed on any devices.

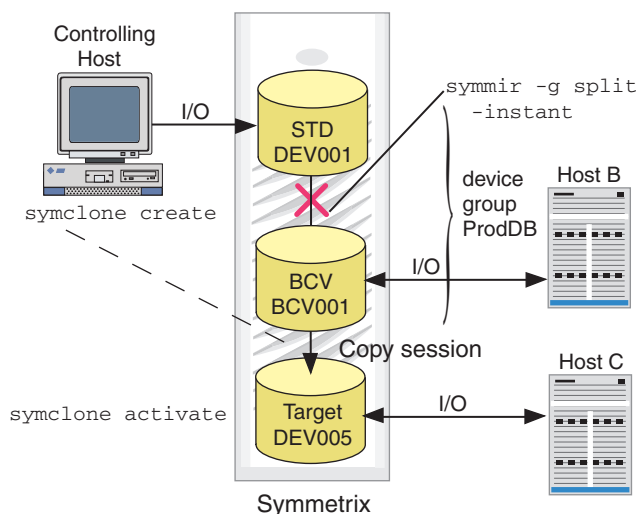


Figure 3-3 Creating a TimeFinder/Clone from a BCV Source

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

The following steps explain the example shown in Figure 3-3, of how to create a CopyOnAccess copy session between a BCV source device and a standard target device:

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 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.

```
symlid -q 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 copies of a standard source device on various targets. A source device can be concurrently copying full copies (-copy option) of same data to up to four target devices at one time. Refer to *Copy Session Limits* on page 3-5 for information about the number of Clone copies that can be created.

To create Clone copies on the four 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 copy sessions can be activated at the same time on the same standard source device.

Standard Device Clone Copies

Figure 3-4 illustrates how to create multiple Clone copies from a standard source device DEV001 on four standard target devices (DEV005, DEV006, DEV007, and DEV008) with various hosts accessing them.

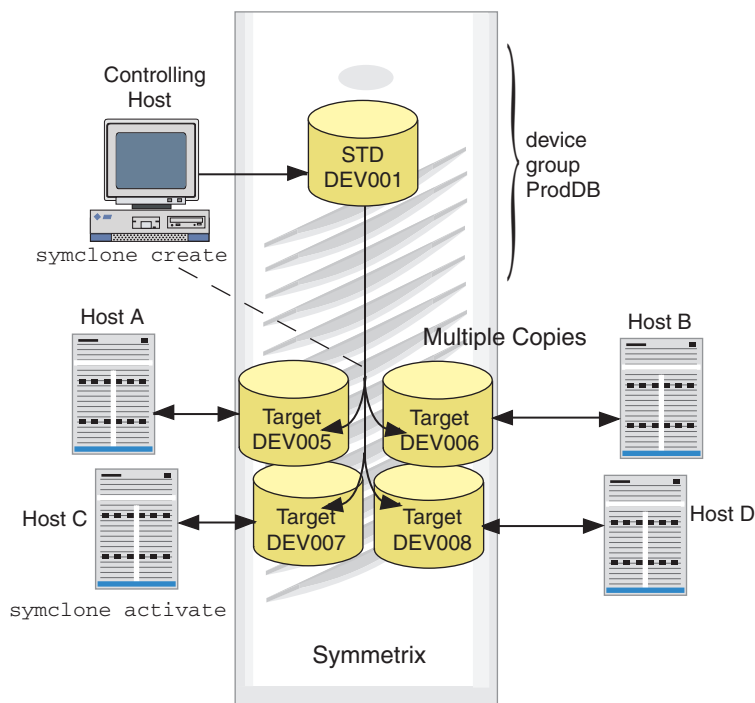


Figure 3-4 Creating Multiple Standard Device Clone Copies

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

The following steps explain the example shown in Figure 3-4, of how to create multiple full copies from a standard source device to four standard target devices:

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\
DEV001 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
```

BCV Device Clone Copies

Figure 3-5 illustrates how to create multiple copies from a BCV source device (DEV001) to four standard target devices (DEV005, DEV006, DEV007, and DEV008) with various hosts accessing them.

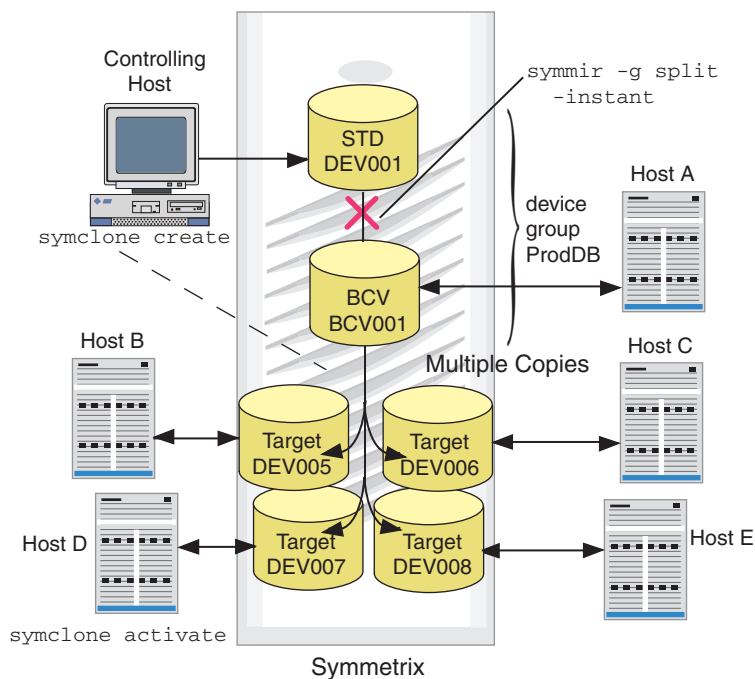


Figure 3-5 Creating Multiple BCV Device Clone Copies

The following steps explain the example shown in Figure 3-5, of how to create multiple full copies from a BCV source device to four standard target devices:

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
```

Note: For additional information on using the `symmir` command and how to perform an instant split operation, refer to Chapter 2.

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 device pair state will be CopyInProgress 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
```

symclone -g Command Line Options

Options to the `symclone -g` command line arguments provide more action flexibility to control copy sessions when you are operating on device(s) of a specified device group.

Table 3-2 lists the `symclone` control operations and the possible options to use when targeting a specified device group.

Table 3-2 symclone -g Control Arguments and Possible Options

Options	Argument Action				
	create	activate	terminate	query	verify
-c, -i	X	X	X	X	X
-consistent		X			
-copied					X
-copy	X				
-copyinprog					X
-copyonaccess					X
-created					X
-exact	X				
-force	X	X	X		
-multi				X	
-noprompt	X	X	X		
-not_ready	X	X	X		
-offline				X	X
-opt	X				
-ppath		X			
-preaction, -postaction		X			
-preservetgtlocks, -lockid	X	X	X		

Table 3-2 **symclone -g Control Arguments and Possible Options (continued)**

Options	Argument Action				
	create	activate	terminate	query	verify
-rdb, -db, -dbtype		X			
-skip	X	X	X		
-symforce			X		
-v	X	X	X		
-vxfs		X			

symclone -file Command Line Options

With the `symclone -file` command, you can perform Clone copy operations on device pairs defined in a device file. The device file includes a source device number and a target device number. You must include the Symmetrix ID in the command line argument.

Table 3-3 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 3-3 symclone -file Control Arguments and Possible Options

Options	Argument Action				
	create	activate	terminate	query	verify
-c, -i	X	X	X	X	X
-consistent		X			
-copied					X
-copy	X				
-copyinprog					X
--copyonaccess					X
-created					X
-force	X	X	X	X	X
-multi				X	
-noprompt	X	X	X		
-not_ready		X	X		
-ppath		X			
-preaction, -postaction		X			
-preservetgtlocks, -lockid	X	X	X		
-rdb, -db, -dbtype		X			
-sid	X	X	X	X	X

Table 3-3 symclone -file Control Arguments and Possible Options (continued)

Options	Argument Action				
	create	activate	terminate	query	verify
-skip	X	X	X		
-symforce			X		
-v	X	X	X		
-vxfs		X			

symclone -cg Command Line Options

Options to the `symclone -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 3-4 lists the `symclone` control operations and the possible options to use when targeting a specified device group.

Table 3-4 symclone -cg Control Arguments and Possible Options

Options	Argument Action				
	create	activate	terminate	query	verify
-c, -i	X	X	X	X	X
-consistent		X			
-copied					X
-copy	X				
-copyinprog					X
-copyonaccess					X
-created					X
-exact	X				
-force	X	X	X		
-multi				X	
-noprompt	X	X	X		
-not_ready		X			
-offline				X	X
-opt	X				
-ppath		X			
-preaction, -postaction		X			
-preservetgtlocks, -lockid	X	X	X		

Table 3-4 **symclone -cg Control Arguments and Possible Options (continued)**

Options	Argument Action				
	create	activate	terminate	query	verify
-rdb, -db, -dbtype		X			
-skip	X	X	X		
-symforce			X		
-v	X	X	X		
-vxfs		X			

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

◆ TimeFinder/Snap Environment	4-2
◆ Creating a Virtual Copy Session	4-5
◆ Activating a Virtual Copy Session.....	4-9
◆ Terminating a Virtual Copy Session.....	4-12
◆ Using a BCV as the Snap Source.....	4-13
◆ Creating Multiple Virtual Copies	4-16
◆ Attaching Source and Target Virtual Devices	4-20
◆ Restoring Data from Virtual Devices	4-21
◆ <code>symsnap -g</code> and <code>-cg</code> Command Line Options	4-29
◆ <code>symsnap -file</code> Command Line Options.....	4-31

TimeFinder/Snap Environment

Since Enginuity Version 5669, SYMCLI TimeFinder/Snap operations allow 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 meta devices can also be created as long as the source and target virtual meta devices 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.

Note: Since Enginuity Version 5568, SYMCLI TimeFinder includes Clone operations that create standard and BCV device clone copies using the `symclone` command. For information about this functionality, refer to Chapter 3.

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 device has associated BCVs, a single source device may have as many as fifteen copies. For more information on the number of available device copies, refer to *Copy Session Limits* on page 4-5.

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 via 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.

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 through using the SYMCLI.

Figure 4-1 illustrates a virtual copy session where the controlling host creates a copy of standard device DEV001 on target device VDEV005.

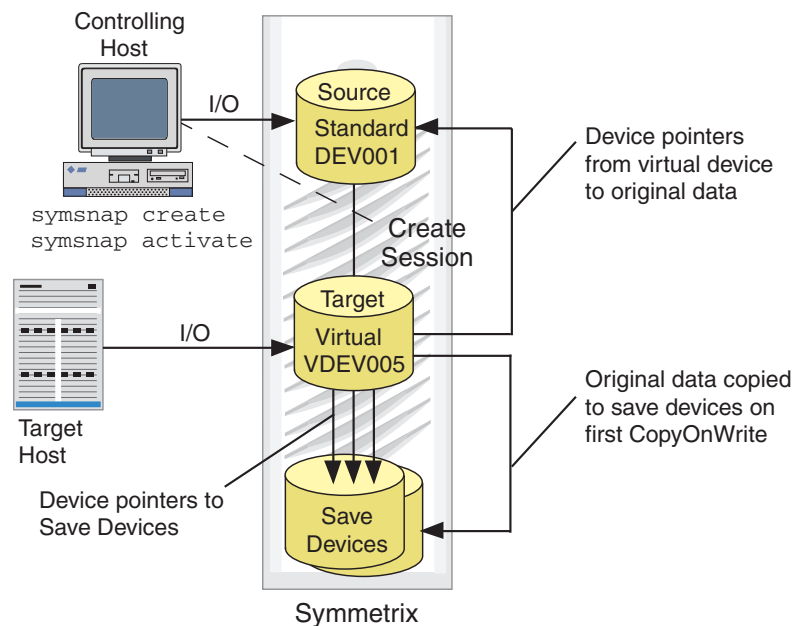


Figure 4-1 Copy of a Standard Device to a Virtual Device (VDEV)

Pair States Ruling Snap Operations

Because various other ongoing operations can conflict with your Snap session, certain rules must be considered. The availability of some Snap copy operations depends on the current state of SRDF and BCV pairs. The following are rules that 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 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 `RestInProgress`.

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 A-9.

For a description of each BCV pair state, refer to *BCV Pair States* in Chapter 2.

Group Operations

TimeFinder/Snap operations support composite groups (`-cg`) or devices within a composite group. For more information about composite groups and device groups, refer to the *Solutions Enabler Symmetrix Base Management CLI Product Guide*.

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 CreateInProgress 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 4-9 for more information.

Note: Depending on the specific level of Engenuity that you are running, the target virtual device cannot be a participant in any other copy session.

Copy Session Limits

The Symmetrix array is currently limited to 15 sessions per source device, which can be used for TimeFinder BCV, Clone, Snap, or SDDF operations. (One session is required for a restore action that would make it 16.) This limits the number of available copies that can be created. A total of 15 concurrent CopyOnWrite copy sessions can be created from a single source device. However, this number is decremented for each current TimeFinder BCV, Clone, Snap, and SDDF session. For example, if you have a standard device with two paired BCV devices, you can only create up to 14 copy sessions for that standard device.

Depending on the specific level of Engenuity 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 4-21.

Copy 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 4-1 on page 4-3 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 if the track has already been marked as unprotected. 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 4-21.

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 3-1 on page 3-7.

Specifying a SAVE Device Pool

Symmetrix SAVE devices are specially configured devices (not mapped to the host) that provide pooled physical storage space used to store pre-update images or changed tracks during a virtual copy session. Devices within a SAVE device pool act as a group to store data in striped form. Symmetrix supports the creation of multiple named SAVE device pools, allowing `symsnap` commands to use a particular pool.

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 4-5 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 snap 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`.

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. When the 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.

SAVE devices can be monitored 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
```

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 Snap 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.

Activating a Virtual Copy Session

To start the Snap copy operation, you must activate the create copy session. To activate the copy session created in the example on page 4-5, 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.

Not Ready Option

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 `symlld ready` command.

PowerPath Option

You can consistently activate multiple virtual copy sessions involving a database using either PowerPath-connected devices or the Engenuity Consistency Assist (ECA) feature. Refer to the *Consistent Option* on page 4-10 for information on using the ECA feature. These features allow Snap copy sessions to be activated with a consistent, restartable copy of the database.

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
```

Consistent Option

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 the Enginuity Consistency Assist (ECA) feature. 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. Refer to Chapter 2 for information on how ECA can be used for consistent split operations.

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. Refer to Figure 4-2 on page 4-11, 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 (see Figure 4-2 on page 4-11), 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.

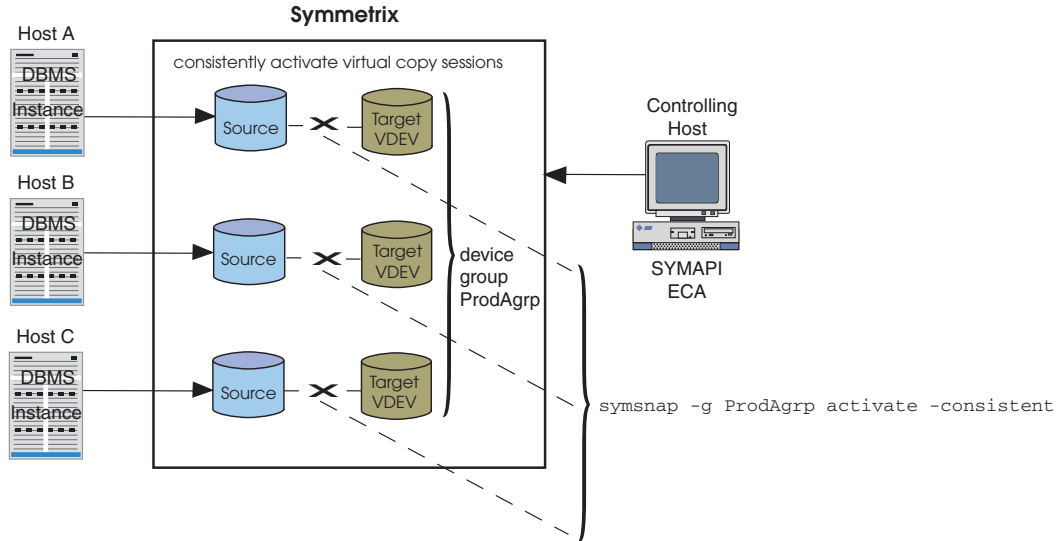


Figure 4-2 TF/Snap Consistent Activate Using ECA

Terminating a Virtual Copy Session

You may need to interrupt an active copy operation at any time. To terminate the copy session activated in the example on page 4-9 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.

Depending upon the specific level of Enginuity Version 5670¹ that you are running, restoring a virtual device may create an additional copy session between the source device and target device of the restore. 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: The `symsnap terminate` command is allowed for all TimeFinder BCV pair states.

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

-
1. Check the *EMC Solutions Enabler Support Matrix, Version 6.0* for specific Enginuity 5670 version levels and restore functionality.

Using a BCV as the Snap Source

As shown in Figure 4-3, 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 on a virtual 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 2.

Note: For this example, with other hosts having access to the BCV source, it might be wise to keep the BCV source Not Ready when you do the initial split.

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: Upon completion of the copy session, you may want to release any Not Ready state you imposed on any devices.

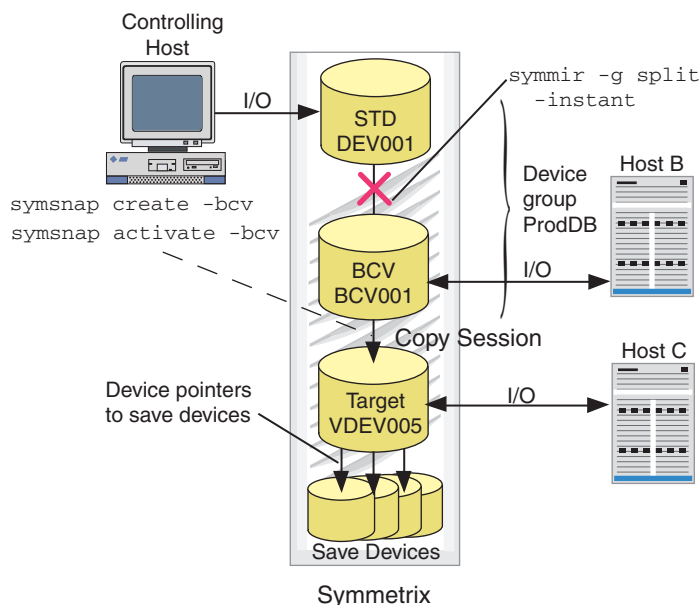


Figure 4-3 Creating a Virtual Copy from a BCV

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

The following steps explain the example shown in Figure 4-3, of how to create a copy session from a BCV source device to a virtual target device:

1. Perform an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it.
2. Verify that the background split is complete. The following command checks every 5 seconds.
3. Create a copy session between the BCV source device BCV001 and the virtual target device VDEV005.

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

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

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

or

```
symmsnap -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
```

Creating Multiple Virtual Copies

You can create up to 15 copies of a source device to various virtual target devices. A source device can be concurrently copying data to up to 15 target devices at one time, depending on the available sessions. Refer to *Copy Session Limits* on page 4-5 for information about the number of copy 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 16 copy sessions can be activated at the same time on the same source device.

Standard Source Device Copies

Figure 4-4 illustrates how to create 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.

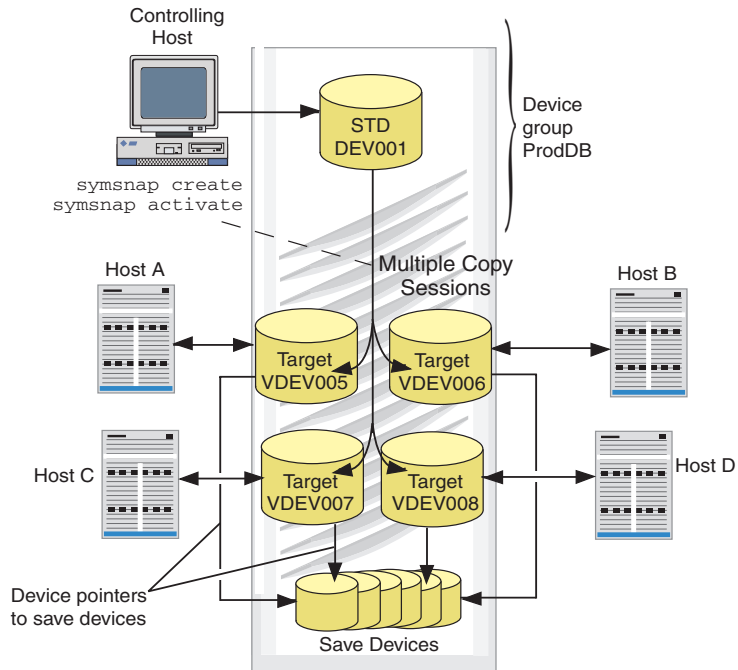


Figure 4-4 Creating Multiple Standard Device Virtual Copies

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 explain the example in Figure 4-4 on page 4-16 of how to create multiple copies from a standard source device to four virtual target devices:

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
```

BCV Source Device Virtual Copies

Figure 4-5 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.

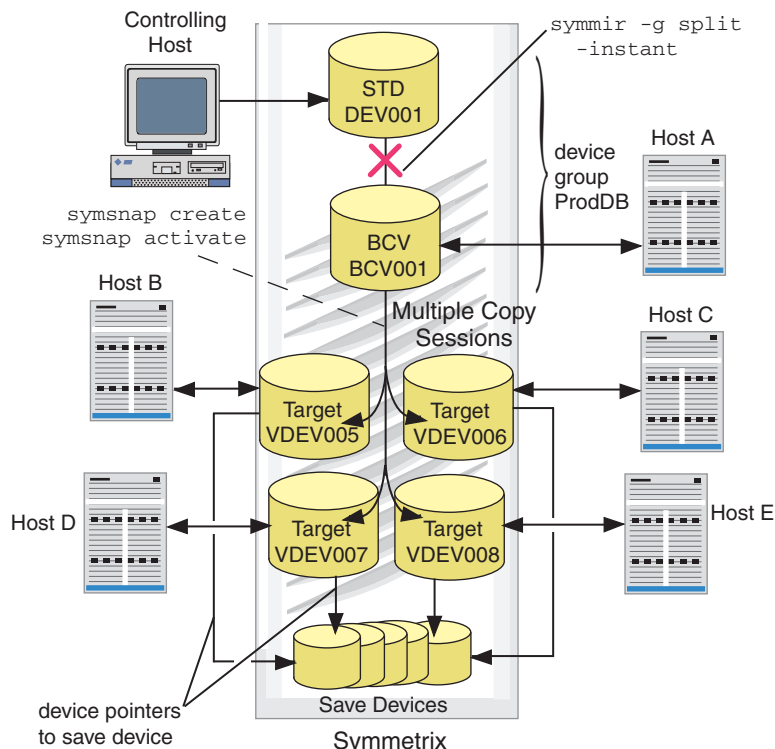


Figure 4-5 Creating Multiple Virtual Copies from a BCV Device

The following steps explain the example in Figure 4-5 of how to create multiple copy sessions from a BCV source device to four virtual target devices:

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
```

Note: For additional information on using the `symmir` command and how to perform an instant split operation, refer to Chapter 2.

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 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 activate BCV001 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
```

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
```

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.

Since Enginuity Version 5670¹, all original Snap copy sessions are maintained when performing a restore operation. 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 *Copy Session Limits* on page 4-5 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.

Note: Depending upon the specific level of Enginuity Version 5670¹ that you are running, the target of the restore operation cannot be a participant in any other Snap or Clone copy session. Also, the original Snap copy session may be automatically terminated after the restore.

1. Check your *EMC Solutions Enabler Support Matrix, Version 6.0*, for specific Enginuity 5670 version levels regarding restore functionality.

Persistent Restore

Since Enginuity Version 5670¹, 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.

Incremental Restore to 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 could 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).

-
1. Check the *EMC Solutions Enabler Support Matrix, Version 6.0*, for specific Enginuity 5670 version levels regarding restore functionality.

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 ld 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 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
```

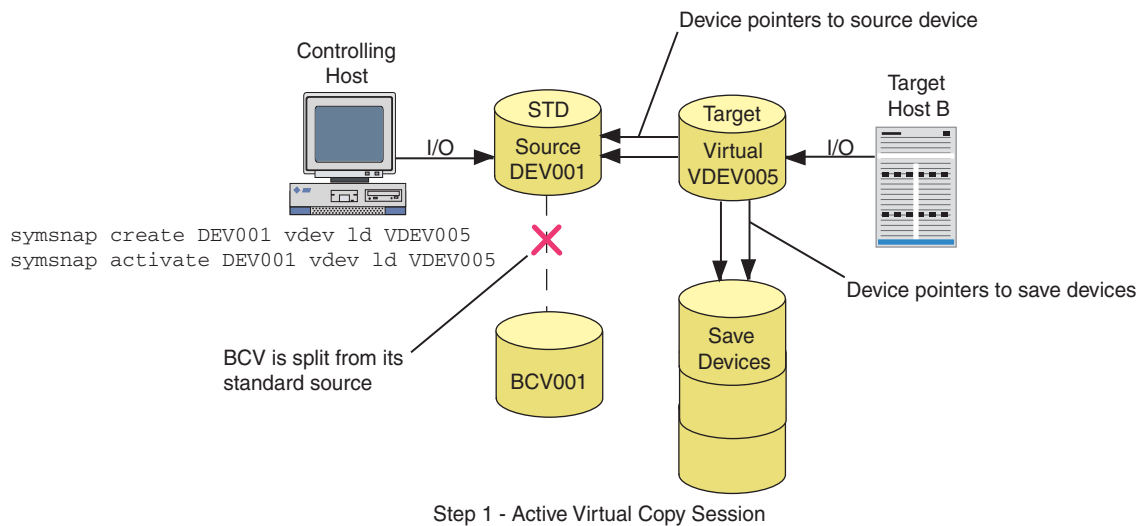
Incremental Restore to a BCV

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

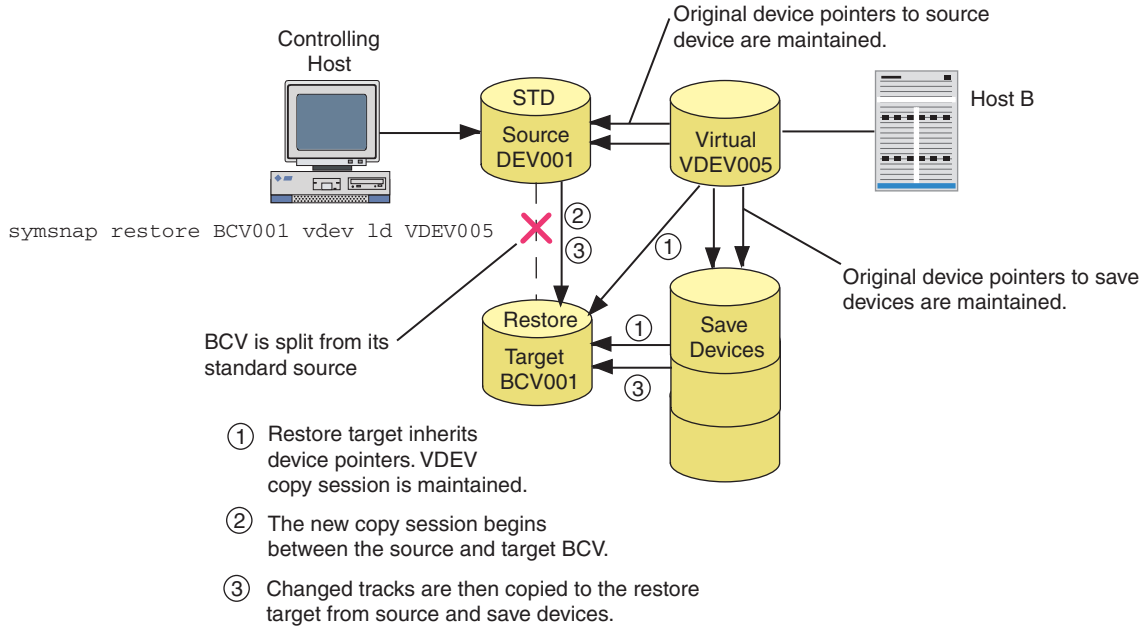
```
symsnap restore BCV001 vdev ld VDEV005
```

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 4-6 on page 4-25. 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). The restore target inherits the virtual device pointers. 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. Once the restore completes, both the original and restore copy sessions are maintained until manually terminated.

Note: Depending upon the specific level of Engenuity Version 5670¹ that you are running, the original virtual device copy session may be automatically terminated and the copy session transferred to the source device and split BCV.



1. Check the *EMC Solutions Enabler Support Matrix, Version 6.0* for specific Engenuity 5670 version levels regarding restore functionality.



Step 2 - Restore the Virtual Copy Session to BCV

Figure 4-6 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 ld VDEV005
symdev -sid SymmID ready 001D
```

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 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 2, *Performing TimeFinder Mirror Operations*.

Full Restore 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
```

Note: Depending upon the specific level of Enginuity Version 5670¹ that you are running, the target of the restore operation cannot be a participant in any other Snap or Clone copy session.

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.

Note: Depending upon the specific level of Enginuity Version 5670¹ that you are running, the original virtual device copy session may be automatically terminated and the Snap copy session transferred to the source device and restore target device.

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.

-
1. Check the *EMC Solutions Enabler Support Matrix, Version 6.0* for specific Enginuity 5670 version levels and restore functionality.

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 ld 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
```

To terminate the restored copy session for a BCV, enter:

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

Note: If restoring to a BCV device, the BCV device must be split from its standard device.

For information on splitting BCV pairs, refer to Chapter 2, *Performing TimeFinder Mirror Operations*.

symsnap -g and -cg Command Line Options

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 4-1 lists the `symsnap` control operations and the possible options to use when targeting a specified device group or composite group.

Table 4-1 symsnap -g and -cg Control Arguments and Possible Options

Options	Argument Action							
	create	activate	terminate	restore	query	verify	attach	detach
-bcv	X	X	X	X				
-c, -i	X	X	X	X	X	X	X	X
-consistent		X						
-copied						X		
-copyonwrite						X		
-created						X		
-exact	X							
-force	X	X	X	X				
-full				X				
-multi					X			
-noprompt	X	X	X	X			X	X
-not_ready		X	X	X				
-offline					X	X		
-ppath		X						
-preaction, -postaction		X						
-preservetgtlocks ^a , -lockid ^a	X	X	X	X				

Table 4-1 `symsnap -g` and `-cg` Control Arguments and Possible Options

Options	Argument Action							
	create	activate	terminate	restore	query	verify	attach	detach
<code>-rdb, -db, -dbtype</code>		X						
<code>-restinprog</code>						X		
<code>-restore</code>					X			
<code>-restored</code>			X			X		
<code>-skip</code>	X	X	X					
<code>-svp</code>	X							
<code>-symforce</code>			X					
<code>-v</code>	X	X	X	X			X	X
<code>-vxfs</code>		X						

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 *Solutions Enabler Symmetrix CLI Command Reference*.

symsnap -file Command Line Options

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 4-2 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 4-2 `symsnap -file` Control Arguments and Possible Options

Options	Argument Action							
	create	activate	terminate	restore	query	verify	attach	detach
-bcv				X				
-c, -i	X	X	X	X	X	X	X	X
-consistent		X						
-copied						X		
-copyonwrite						X		
-created						X		
-force	X	X	X	X				
-full				X				
-multi					X			
-noprompt	X	X	X	X				
-not_ready		X	X	X				
-ppath		X						
-preaction, -postaction		X						
-preservetgtlocks, -lockid	X	X	X	X				
-rdb, -db, -dbtype		X						
-restinprog						X		

Table 4-2 `symsnap -file` Control Arguments and Possible Options (continued)

Options	Argument Action							
	create	activate	terminate	restore	query	verify	attach	detach
-restore					X			
-restored			X			X		
-sid	X	X	X	X	X	X	X	X
-skip	X	X	X					
-symforce			X					
-v	X	X	X					
-vxfs		X						

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

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 A-2
- ◆ State Rules for TimeFinder/Clone Operations A-4
- ◆ State Rules for TimeFinder/Snap Operations A-9

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 A-1 provides a description the various SRDF pair states.

Table A-1 SRDF Pair States

State	Description
SyncInProg	A synchronization is currently in progress between the R1 and the R2. There are existing invalid tracks between the two pairs and the logical link between both sides of an RDF pair is up.
Synchronized	The R1 and the R2 are currently in a synchronized state. The same content exists on the R2 as the R1. There are no invalid tracks between the two pairs.
Split	The R1 and the R2 are currently Ready to their hosts, but the link is Not Ready or write disabled.
Failed Over	The R1 is currently Not Ready or write disabled and operations have been failed over to the R2.
R1 Updated	The R1 is currently Not Ready or write disabled to the host, there are no local invalid tracks on the R1 side, and the link is Ready or write disabled.
R1 UpdInProg	The R1 is currently Not Ready or write disabled to the host, there are invalid local (R1) tracks on the source side, and the link is Ready or write disabled.
Suspended	The RDF links have been suspended and are Not Ready or write disabled. If the R1 is Ready while the links are suspended, any I/O will accumulate as invalid tracks owed to the R2.
Partitioned	The SYMAPI is currently unable to communicate through the corresponding RDF path to the remote Symmetrix 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.
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.

Table A-1 SRDF Pair States (continued)

State	Description
Mixed	A composite SYMAPI device group RDF pair state. There exists different SRDF pair states within a device group.
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.

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.

Sync In Progress

Table A-2 identifies the `symclone` copy operations available for use when SRDF pairs are in the SyncInProgress pair state.

Table A-2 TimeFinder/Clone Operations for SyncInProgress Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X ^{a, c, e}	X ^d	
Activate	X ^a	X ^{a, c, e}	X ^{b, d}	
Terminate	X	X	X	X

Synchronized

Table A-3 identifies the `symclone` copy operations available for use when SRDF pairs are in the Synchronized pair state.

Table A-3 TimeFinder/Clone Operations for Synchronized Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X ^{c, e}	X ^d	X
Activate	X	X ^{c, e}	X ^d	X
Terminate	X	X	X	X

- 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed with CopyOnAccess.

Split

Table A-4 identifies the `symclone` copy operations available for use when SRDF pairs are in the Split pair state.

Table A-4 TimeFinder/Clone Operations for Split Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X	X	X ^{d, f}
Activate	X	X	X	X ^{d, f}
Terminate	X	X	X	X

Failed Over

Table A-5 identifies the `symclone` copy operations available for use when SRDF pairs are in the Failed Over pair state.

Table A-5 TimeFinder/Clone Operations for Failed Over Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X		X ^d	X ^{d, f}
Activate	X		X ^d	X ^{d, f}
Terminate	X	X	X	X

- Action is not allowed if there are local R1 invalids or remote R2 invalids.
- Action is not allowed if there are remote R1 invalids or local R2 invalids.
- The force option (`-force`) must be applied.
- Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.
- Action is not allowed with CopyOnAccess.
- Action is not allowed if the target device is an R2 larger than the R1.

R1 Updated

Table A-6 identifies the `symclone` copy operations available for use when SRDF pairs are in the R1 Updated pair state.

Table A-6 TimeFinder/Clone Operations for R1 Updated Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X		X ^d	
Activate	X ^g		X ^d	
Terminate	X	X	X	X

R1 Update In Progress

Table A-7 identifies the `symclone` copy operations available for use when SRDF pairs are in the R1 UpdInProg pair state.

Table A-7 TimeFinder/Clone Operations for R1 UpdInProg Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create			X ^d	
Activate	X ^g		X ^d	
Terminate	X	X	X	X

- Action is not allowed if there are local R1 invalids or remote R2 invalids.
- Action is not allowed if there are remote R1 invalids or local R2 invalids.
- The force option (`-force`) must be applied.
- Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.
- Action is not allowed with CopyOnAccess.
- Action is not allowed if the target device is an R2 larger than the R1.
- Action is not allowed if the `-consistent` option was specified.

Suspended

Table A-8 identifies the `symclone` copy operations available for use when SRDF pairs are in the Suspended pair state.

Table A-8 TimeFinder/Clone Operations for Suspended Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X	X	X ^{d, f}
Activate	X	X	X	X ^{d, f}
Terminate	X	X	X	X

Partitioned

Table A-9 identifies the `symclone` copy operations available for use when SRDF pairs are in the Partitioned pair state.

Table A-9 TimeFinder/Clone Operations for Partitioned Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X	X ^d	X ^{d, f}
Activate	X	X	X ^d	X ^{d, f}
Terminate	X	X	X	X

- Action is not allowed if there are local R1 invalids or remote R2 invalids.
- Action is not allowed if there are remote R1 invalids or local R2 invalids.
- The force option (`-force`) must be applied.
- Action is not allowed if the R2 source or target is in SRDF/Asynchronous mode.
- Action is not allowed with `CopyOnAccess`.
- Action is not allowed if the target device is an R2 larger than the R1.
- Action is not allowed if the `-consistent` option was specified.

Consistent

Table A-10 identifies the `symclone` copy operations available for use when SRDF pairs are in the Consistent pair state for asynchronous mode.

Table A-10 TimeFinder/Clone Operations for Consistent Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create	X	X ^{a, c, e}		
Activate	X ^a	X ^{a, c, e}		
Terminate	X	X	X	X

Invalid

Table A-11 identifies the `symclone` copy operations available for use when SRDF pairs are in the Invalid pair state.

Table A-11 TimeFinder/Clone Operations for Invalid Pair States

TimeFinder/Clone Action	R1 Copy Source	R1 Copy Target	R2 Copy Source	R2 Copy Target
Create				
Activate				
Terminate	X	X	X	X

- 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed with CopyOnAccess.
- f. Action is not allowed if the target device is an R2 larger than the R1.
- g. Action is not allowed if the `-consistent` option was specified.

State Rules for TimeFinder/Snap Operations

This section identifies what `symsnap` copy actions are available for use within each of the SRDF pair states.

Sync In Progress

Table A-12 identifies the `symsnap` copy operations available for use when SRDF pairs are in the SyncInProgress pair state.

Table A-12 TimeFinder/Snap Operations for SyncInProgress Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X ^a	N/A	X ^{b, d}	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X ^{a, c}	N/A		N/A
Incremental restore to a split BCV or full restore to any device	X	X ^{a, c}	X ^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.

Synchronized

Table A-13 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Synchronized pair state.

Table A-13 TimeFinder/Snap Operations for Synchronized Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X ^c	N/A		N/A
Incremental restore to a split BCV or full restore to any device	X	X ^c	X ^d	

Split

Table A-4 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Split pair state.

Table A-14 TimeFinder/Snap Operations for Split Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X	N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X	X	X ^d	X ^{d, f}

- 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed if the target device is an R2 larger than the R1.

Failed Over

Table A-15 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Failed Over pair state.

Table A-15 TimeFinder/Snap Operations for Failed Over Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X	N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X		X ^d	X ^{d, f}

R1 Updated

Table A-16 identifies the `symsnap` copy operations available for use when SRDF pairs are in the R1 Updated pair state.

Table A-16 TimeFinder/Snap Operations for R1 Updated Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X ^g	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source		N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X ^g		X ^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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed if the target device is an R2 larger than the R1.
- f. Action is not allowed if the `-consistent` option was specified.

R1 Update In Progress

Table A-17 identifies the `symsnap` copy operations available for use when SRDF pairs are in the R1 UpdInProg pair state.

Table A-17 TimeFinder/Snap Operations for R1 UpdInProg Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X ^g	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source		N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X ^g		X ^d	

Suspended

Table A-18 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Suspended pair state.

Table A-18 TimeFinder/Snap Operations for Suspended Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X	N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X	X	X ^d	X ^{d, e}

- 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed if the target device is an R2 larger than the R1.
- f. Action is not allowed if the `-consistent` option was specified.

Partitioned

Table A-19 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Partitioned pair state.

Table A-19 TimeFinder/Snap Operations for Partitioned Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A	X ^d	N/A
Activate	X	N/A	X ^d	N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X	N/A	X ^d	N/A
Incremental restore to a split BCV or full restore to any device	X	X	X ^d	X ^{d, f}

Invalid

Table A-20 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Invalid pair state.

Table A-20 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	X	N/A	X	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 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed if the target device is an R2 larger than the R1.
- f. Action is not allowed if the `-consistent` option was specified.

Consistent

Table A-21 identifies the `symsnap` copy operations available for use when SRDF pairs are in the Consistent pair state.

Table A-21 TimeFinder/Snap Operations for Consistent Pair States

TimeFinder/Snap Action	R1 Snap Source	R1 Snap Target	R2 Snap Source	R2 Snap Target
Create	X	N/A		N/A
Activate	X ^a	N/A		N/A
Terminate	X	N/A	X	N/A
Incremental restore to source	X ^{a, c}	N/A		N/A
Incremental restore to a split BCV or full restore to any device	X	X ^{a, c}		

- 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 if the R2 source or target is in SRDF/Asynchronous mode.
- e. Action is not allowed if the target device is an R2 larger than the R1.
- f. Action is not allowed if the `-consistent` option was specified.

**TimeFinder State
Reference**

This appendix describes the applicable TimeFinder pair states that rule TimeFinder/Clone and TimeFinder/Snap copy session operations.

- ◆ TimeFinder Pair States..... B-2
- ◆ Pair States Ruling TimeFinder/Clone Operations..... B-3
- ◆ Pair States Ruling TimeFinder/Snap Operations..... B-4

TimeFinder Pair States

Certain TimeFinder/Clone and TimeFinder/Snap copy operations are not allowed within Symmetrix storage 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 B-1 provides a description the various TimeFinder pair states.

Table B-1 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

Tables B-2 identifies which `symclone` copy **create** actions are available for use for each of the TimeFinder pair states.

Table B-2 TimeFinder/Clone Operations for Pair States - CreateAction

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 B-3 identifies which `symsnap` copy **create** actions are available for use for each of the TimeFinder pair states.

Table B-3 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 B-4 identifies which `symsnap` copy **restore** actions are available for use for each of the TimeFinder pair states.

Table B-4 TimeFinder Snap Operations for Pair States - RestoreAction

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.

Numerics

2-way mirror split 2-60

B

BCV control operations

full establish 2-10, 2-28

full restore 2-11, 2-72

incremental establish 2-10, 2-40

incremental restore 2-11, 2-79

split 2-11, 2-47

BCV devices 2-2

associating all 2-16

device groups 2-15

disassociating 2-21

list 2-14

moving all 2-23

remote configurations 2-119

BCV mirror 2-7

BCV pair 2-27

mirrors in a 2-27

query state of 2-96

verify 2-97

BCV pair states 2-99

Never Established 2-99

NeverEstab 2-99

RestInProg 2-99, B-2

Restore In Progress 2-99, B-2

Restored 2-99, B-2

Split 2-99, B-2

Split Before Restore 2-99, B-2

Split Before Sync 2-99, B-2

Split In Progress 2-99, B-2

Split No Incremental 2-99, B-2

SplitBfrRest 2-99, B-2

SplitBfrSync 2-99, B-2

SplitInProg 2-99, B-2

SplitNoInc 2-99, B-2

Synchronized 2-99, B-2

SyncInProg 2-99, B-2

BCV pairs 2-7

establishing 2-27

preferred attachment 2-87

restoring 2-79

splitting 2-47

BCV states 2-100

both sides 2-67

business continuance 2-2

business continuance operations 2-2

C

clone copies 3-2

Clone emulation 2-12

commands, multi-hop RDF operations 2-122

comments xvi

concurrent BCV Pairs

establishing 2-36

splitting 2-58

consistent split 2-64

both sides 2-67

Copied device state 3-6

copy session 4-5

copy session limits 3-5, 4-5

CopyInProgress device state 3-6

CopyOnAccess device state 3-6

CopyOnWrite device state 4-6

Created device state 3-5, 4-5

CreateInProgress device state 3-5, 4-5

D

database locking 2-104

databases

freezing 2-61

I/O controls 2-61

thawing 2-61

device copy

full 3-6

device external locks 4-4

device file

for symclone command 3-25

for symmir command 2-105

for symsnap command 4-31

using with the symmir command 2-116

device label ambiguity 2-47, 2-127

device states

Copied 3-6

CopyInProgress 3-6

CopyOnAccess 3-6

CopyOnWrite 4-6

Created 3-5

CreateInProgress 3-5

differential split 2-53

E

Enginuity Consistency Assist

creating consistent clone copies 3-10

performing consistent split operations 2-68

Enterprise TimeFinder consistent split 2-64

establish command

using to create and activate 3-8

using to recreate and activate 3-12

external locks, device 2-8

F

file, devices 2-105, 3-25, 4-31

force on operations 2-106

full restore 2-73

H

hot backup 2-62

I

incremental establish 2-40

incremental restore 2-79

instant BCV mirror sync 2-38

instant split 2-54

instant/consistent split 2-64

L

labels, devices 2-127

locks, device external 2-8

M

Meta devices 2-2

moveall

restore 2-86, 2-108

splits 2-60

moving mirror 2-56

multi-hop

configurations 2-119

control operations 2-122

mirroring 2-6

multiple BCV pairs 2-42, 2-82

full establishing 2-30

O

offline option 2-108

options to

symmir -cg 2-114

symmir -f 2-116

symmir -file 2-117

symmir -g 2-101, 2-102, 2-113

P

pairing algorithm 2-39

persistent restore 4-22

PowerPath 4-9

protected BCV establish 2-38

protected restore 2-86, 2-108

Q

Query BCV pairs 2-96

R

- RAID-1 2-2, 2-3
- RAID-S 2-2, 2-3
- relabel operation 2-127
- remote BCV devices 2-119
- remote BCV pairs
 - establishes 2-33, 2-35
 - incremental establishes 2-43
 - incremental restores 2-83
 - restores 2-75
 - split 2-50
- remote operations 2-119
- reverse split 2-56

S

- SAVE devices 1-4
 - monitoring usage 4-7
 - specifying 4-7
 - specifying a device pool 4-7
- split, moveall 2-60
- splits, Enterprise TimeFinder consistent 2-64
- SRDF pair states A-2
- standard devices 2-2
- states, BCV operations 2-100
- SYMCLI commands
 - symclone 3-2
 - symioctl 2-61
 - symmir 1-7, 2-10
- symclone command options
 - device file 3-25
- Symmetrix Remote Data Facility (SRDF) 2-4
- symmir command options
 - attached 2-112
 - background 2-112
 - BCV mirror 2-111
 - both sides 2-103
 - bypass 2-104
 - concurrent 2-111
 - count option 2-104
 - DB 2-105
 - DB type 2-105
 - device file 2-105, 2-116
 - exact 2-106
 - force 2-106
 - instant 2-107

- interval 2-104
- multi-BCV 2-112
- no prompt 2-107
- not ready 2-108
- offline 2-108
- optimize 2-107
- optimize remote repairing 2-115
- post-action 2-108
- PowerPath 2-108
- pre-action 2-108
- RDB 2-105
- RDF option 2-109
- remote 2-109, 2-125
- restinprog state 2-111
- restored 2-111
- reverse option 2-110
- sid 2-110
- split state 2-111
- symforce 2-110
- synchronized state 2-103, 2-105, 2-111
- syncinprog state 2-111
- verbose 2-110
- VERITAS FS 2-110
 - wait for background split 2-54
- symsnap 4-3, A-4, A-5, A-6, A-7, A-8, A-9, A-10, A-11, A-12, A-13, B-4
- symsnap command options
 - device file 4-31
- synchronization, confining 2-9
- synchronized BCV pair 2-41

T

- TimeFinder BCV operations 2-8
- TimeFinder consistent split
 - both sides 2-67
- TimeFinder/Clone emulation 2-12
- TimeFinder/Clone operations 3-2
- TimeFinder/Mirror control operations 2-10
- TimeFinder/Snap operations 4-2
- transient BCV pair states 2-100

V

- virtual devices 1-3
 - restoring 4-21

