

**EMC Solutions Enabler
Symmetrix Configuration Change CLI
Version 6.0**

PRODUCT GUIDE

**P/N 300-000-882
REV A07**

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

Copyright © 2002, 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	xi
----------------------	----

Chapter 1 Overview

Introduction	1-2
Symmetrix Device Types	1-4
Dynamic RDF Devices	1-4
Device Masking (VCM) Devices.....	1-4
DRV Devices	1-4
Meta Devices	1-6
Virtual Devices	1-9
SAVE Devices	1-9
Devices and System Configurations	1-10
Configurations	1-10
SRDF and BCV Configurations	1-15
Command Summary	1-17

Chapter 2 Performing Configuration Operations

Configuration Control Operations Overview.....	2-2
Device Locking.....	2-3
Held Devices	2-3
Operations Classes	2-3
Configuration Change Safety.....	2-5
The Command File	2-7
Device Creation Strategy	2-8
Device Creation Restrictions	2-9
Meta Configuration Restrictions	2-11
Virtual Device Configuration Restrictions	2-11
SAVE Device Configuration Restrictions	2-11

RDF Configuration Changes.....	2-12
Changes to Mirror Protection	2-12
Creating Devices	2-13
Deleting Devices	2-17
Converting Devices	2-19
Setting Device Characteristics.....	2-22
Mapping Devices	2-24
Unmapping Devices.....	2-26
Configuring Dynamic RDF-Capable Devices.....	2-28
Enabling Dynamic RDF on the Symmetrix.....	2-29
Enabling Dynamic RDF on the Device.....	2-30
Setting RDF Group Attributes	2-31
Configuring Meta Devices.....	2-32
Creating a Meta Device	2-32
Expanding an Existing Meta Device.....	2-35
Removing Meta Members	2-36
Dissolving a Meta Device.....	2-36
Converting a Meta Device.....	2-36
Data Preservation Concerns.....	2-37
Managing CKD Devices	2-39
Mapping a Range of Devices	2-39
Unmapping a Range of Devices	2-40
Copying the Mapping of a Range of Devices.....	2-40
Assigning PAV Alias Addresses to Mapped Devices.....	2-41
Removing PAV Alias Addresses From Mapped Devices	2-41
Managing SAVE Device Pools	2-42
Creating a SAVE Device Pool.....	2-42
Deleting a SAVE Device Pool.....	2-42
Adding SAVE Devices to a Pool.....	2-43
Removing SAVE Devices From a Pool	2-43
Enabling/Disabling SAVE Devices.....	2-44
Setting Port Characteristics	2-45
Setting Symmetrix Metrics	2-50
Swapping RA Groups	2-52
Managing Spare Disks	2-54
Adding a Spare Disk	2-54
Deleting a Spare Disk.....	2-55
Viewing Existing Spares.....	2-55
Swapping R1 Devices with R2 Devices	2-56
Stages of the Configuration Change Action	2-58
Monitoring the Status of the Change Session.....	2-61
Aborting the Change Session.....	2-62
Subsystem IDs for Mainframe Systems.....	2-63

Appendix A Updating the Host

Sun Solaris Systems A-2

HP-UX Systems A-3

IBM AIX Systems A-4

HP Tru64 UNIX Systems A-5

Windows Systems A-6

 Windows 2000..... A-6

 Windows 2003..... A-6

Index i-1

1-1	Volume Stages of a DRV Swap	1-5
1-2	Concatenated Meta Devices	1-6
1-3	Striped Data	1-7
1-4	Symmetrix Concatenated Meta Device Two-Way Mirror Scheme	1-12
1-5	Symmetrix Three-Way Mirror Scheme	1-13
1-6	Symmetrix Parity RAID Mirror Scheme	1-14
1-7	Simple SRDF Device Configuration	1-15
1-8	Various SRDF-Connected Device Configurations with BCVs	1-16

1-1	Meta Volume Stripe Sizes	1-8
1-2	Mirror Set Protection Schemes	1-10
1-3	Symconfigure Commands Summary	1-17
2-1	Steps to Create a Device	2-8
2-2	Steps to Create a Meta Device from One Type to Another	2-10
2-3	Meta Device Data Preservation	2-38
2-4	SCSI Protocol Port Flags	2-46
2-5	Fibre Protocol Port Flags	2-48

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 a 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 advanced command-line users and script programmers by describing how to manage various types of configuration changes on Symmetrix devices using the SYMCLI commands of the EMC Solutions Enabler software.

Organization The following defines the structure of this guide:

Chapter 1, *Overview*, introduces the Solutions Enabler Configuration Component and describes the types of devices and configurations.

Chapter 2, *Performing Configuration Operations*, describes how to perform control operations that manage various types of configurations using SYMCLI.

Appendix A, *Updating the Host*, describes how to update your host so that the host recognizes the new Symmetrix configuration.

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 Installation Guide*
- ◆ *EMC Solutions Enabler SRDF Family CLI Product Guide,*
- ◆ *EMC Solutions Enabler Symmetrix SRM CLI Product Guide*
- ◆ *EMC Solutions Enabler TimeFinder Family CLI Product Guide*
- ◆ EMC Host Connectivity Guides for [your operating system]

The following are EMC Engineering white papers of related interest:

- ◆ *Using the SYMCLI Configuration Manager*
- ◆ *Using the SYMCLI Configuration Manager for Managing CKD Device*

Note: 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*.

Conventions Used in this Manual

The following conventions are used in this manual:

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

Every occurrence of the word MVS in Solutions Enabler software symbols, means OS/390 and z/OS.

Every occurrence of *Symmetrix 4*, refers to Symmetrix model numbers: 3330, 3430, 3630, 3700, 3830, 3930, 5330, 5430, 5630, 5700, 5830, and 5930.

Every occurrence of *Symmetrix 5*, refers to Symmetrix model numbers: 8130, 8230, 8430, 5830, 8730, and 8830.

Every occurrence of *Symmetrix DMX*, refers to Symmetrix model numbers: DMX 800, DMX 1000, DMX 2000, DMX 1000-P, and DMX 2000-P.

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>	<i>Italic text and characters emphasizes new terms, identifies variables in a software syntax (non-literal notation), identifies unique word usage, and applies emphasis in examples and in references to book titles and sections.</i>
<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

For questions about technical support, call your local sales office or service provider.

If you have a valid EMC service contract, contact EMC Customer Service at:

- United States:** (800) 782-4362 (SVC-4EMC)
- Canada:** (800) 543-4782 (543-4SVC)
- Worldwide:** (508) 497-7901

Follow the voice menu prompts to open a service call and select the applicable product support.

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.

This chapter introduces the Solutions Enabler Configuration Component and describes the types of devices and configurations.

- ◆ Introduction 1-2
- ◆ Symmetrix Device Types..... 1-4
- ◆ Devices and System Configurations 1-10
- ◆ Command Summary 1-17

Introduction

The SYMCLI Configuration Component extends the basic SYMCLI command set to include the `symconfigure` command on the following platforms:

- ◆ Sun Solaris
- ◆ Hewlett Packard HP-UX
- ◆ Windows 2000
- ◆ Tru64 UNIX
- ◆ IBM AIX
- ◆ Linux
- ◆ AS/400 (via client/server)
- ◆ Linux S/390

This command allows you to perform powerful control operations that manage various types of configuration changes on devices in your Symmetrix® environment. You can invoke `symconfigure` from the local host to make configuration changes to a locally connected Symmetrix array, or to an RDF-linked Symmetrix array.

The `symconfigure` command enables you to manage change sessions with the following control operations:

- ◆ Creates new Symmetrix devices
- ◆ Creates/deletes SAVE device pools
- ◆ Converts devices to other configuration types by adding/removing RDF, BCV, or DRV attributes
- ◆ Copies device mappings from one EA/EF port to another
- ◆ Increases device protection by adding additional mirrors
- ◆ Decreases device protection by removing existing mirrors
- ◆ Deletes devices
- ◆ Disables SRDF/A on an RA group
- ◆ Sets device attributes (for example, WORM)
- ◆ Sets selective Symmetrix metrics (for example, parity RAID)
- ◆ Swaps RDF source/target attributes for an RA group
- ◆ Converts static RDF devices to dynamic RDF devices
- ◆ Forms/dissolves meta devices
- ◆ Adds/removes meta device members
- ◆ Maps/unmaps a range of Symmetrix devices that form a mainframe CU image to a front-end port
- ◆ Maps/unmaps a Symmetrix device to front-end ports
- ◆ Sets/resets SCSI or fibre port flags
- ◆ Adds/removes dynamic (hot) spare
- ◆ Adds/removes PAV aliases from mainframe 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*.

Symmetrix Device Types

This section describes the Symmetrix device types that can apply to the various possible configurations.

Dynamic RDF Devices

Since Enginuity™ Version 5568, devices can be configured to be Dynamic RDF-capable devices. Dynamic RDF functionality enables you to create, delete, and swap SRDF® pairs while the Symmetrix array is in operation. Using Dynamic RDF technology, you can establish SRDF device pairs from non-SRDF devices, and then synchronize and manage them in the same way as configured SRDF pairs.

The Dynamic RDF configuration state of the Symmetrix array must be enabled in SymmWin or via the Configuration Manager and the devices must be designated as dynamic RDF-capable devices.

For information about dynamic RDF devices, refer to the *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.

Device Masking (VCM) Devices

Symmetrix device masking or VCM devices are Symmetrix devices that have been masked for visibility to certain hosts. The device masking database (VCMDB), which holds device masking records, typically resides on a small disk device (such as a 48-cylinder, 24 MB device). For more information, see the *EMC Solutions Enabler Symmetrix Device Masking CLI Product Guide*.

DRV Devices

A *Dynamic Reallocation Volume* (DRV) is a nonuser addressable Symmetrix device used by the Symmetrix Optimizer to temporarily hold user data while reorganization of the devices is being executed. Typically, it is used by the Optimizer in logical volume swapping operations. Figure 1-1 on page 1-5 illustrates the stages and status of the volumes during a complete full volume swap (vol 1 with 4).

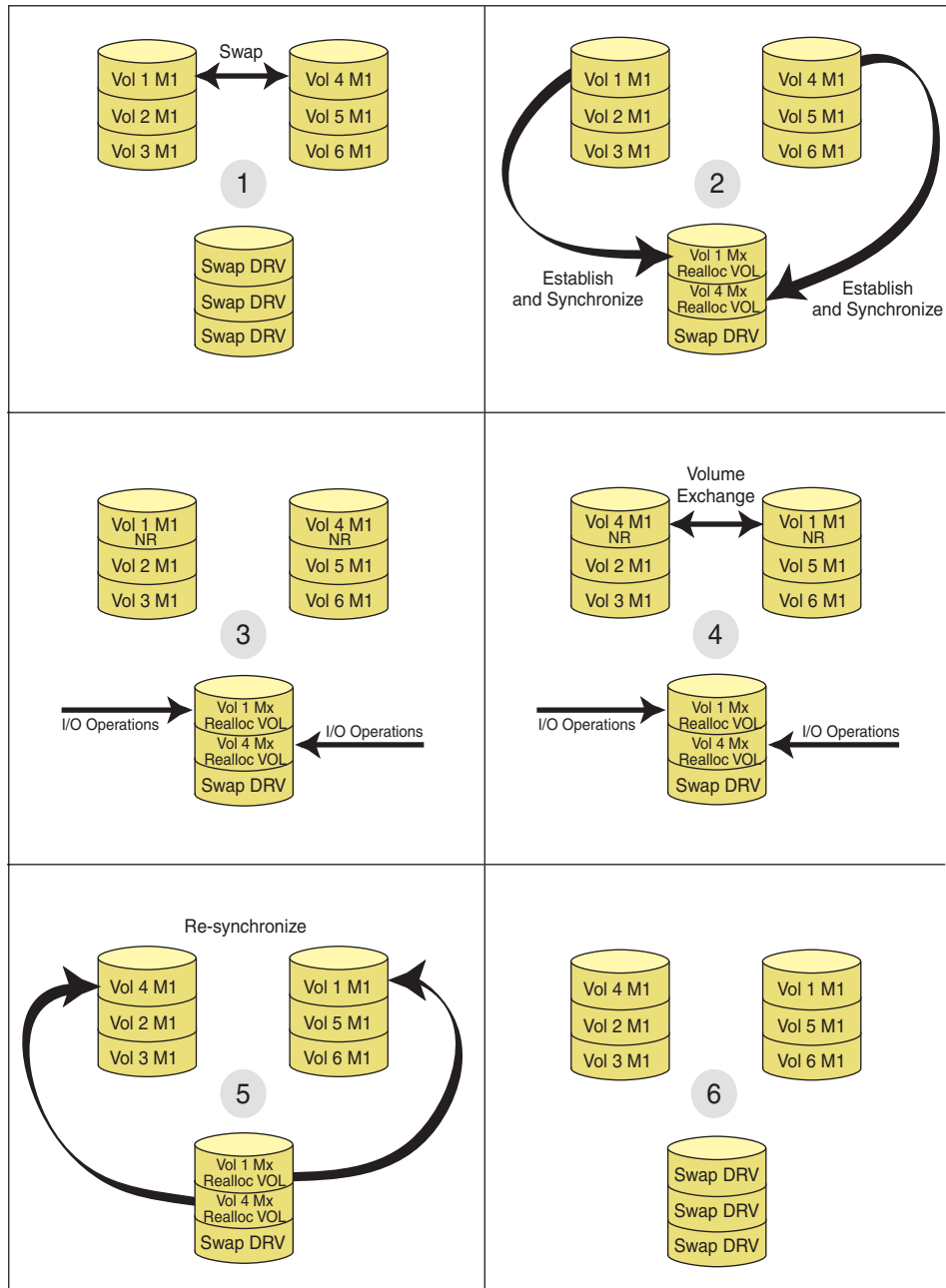


Figure 1-1 Volume Stages of a DRV Swap

Meta Devices

Meta devices allow individual devices to be concatenated to create larger devices. For Enginuity Versions below 5x66, meta devices must be adjacent Symmetrix devices and are formed by grouping physical devices that followed each other. For Enginuity Version 5x66 and higher, the devices assigned in a meta sequence do not need to be adjacent.

The *meta head* is the first device in the meta device and is responsible for receiving incoming commands. When an incoming command for the meta head is processed, the Enginuity software determines which meta device member should execute the command.

Several operating systems, such as Windows 2000 and some open systems environments, can have larger volumes than are provided by standard Symmetrix physical disk devices. A meta device (volume) is a logical volume set created from individual Symmetrix hypervisors to define volumes larger than the current Symmetrix maximum hyper volume size of approximately 16 GB. Meta volumes (devices) are functionally the same as logical volume sets implemented with the host volume manager software. Physically, a meta device is two or more Symmetrix hyper volumes presented to a host as a single addressable device. It consists of a meta head device, some number of member devices (optional), and a meta tail device (see Figure 1-2).

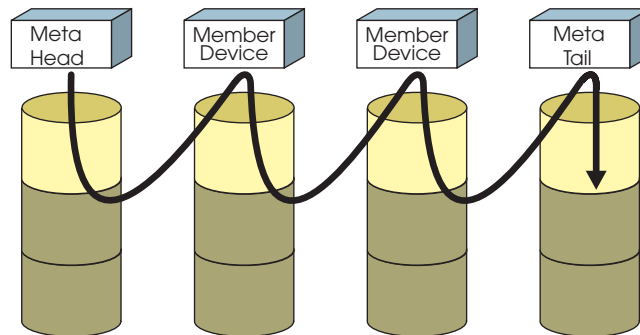


Figure 1-2 Concatenated Meta Devices

Meta Device Size Requirements

Meta devices may contain as many as 255 members. The maximum size meta device that has been tested by EMC is 3.825 terabytes, which is the maximum size supported at this time. In Enginuity Version 5x66 and higher, Meta devices can consist of nonsequential and nonadjacent volumes.

Meta Performance

By allowing individual physical disk devices to be grouped together into a meta device (volume) and the capability of meta device addressing, Symmetrix enhances disk system functionality. To increase throughput and further improve performance, Symmetrix provides multiple I/O drive queues for meta volumes.

Accessing Data in a Meta

Addressing of data contained in a meta device can be organized in two different ways:

- ◆ Concatenated devices
- ◆ Striped devices

Concatenated Devices

Concatenated devices are volume sets that are organized with the first byte of data at the beginning of the first device (Figure 1-2 on page 1-6). Addressing continues to the end of the first device before any data on the next device is referenced. When writing to a concatenated device, the first meta device member receives all the data until it is full, and then data is directed to the next member and so on.

Striped Data

Meta device addressing by striping divides each meta member device into a series of stripes, addressing a stripe from each device before advancing to the next stripe on the first device (as shown in Figure 1-3). When writing to a striped volume, equal size stripes of data from each participating drive are written alternately to each member of the set.

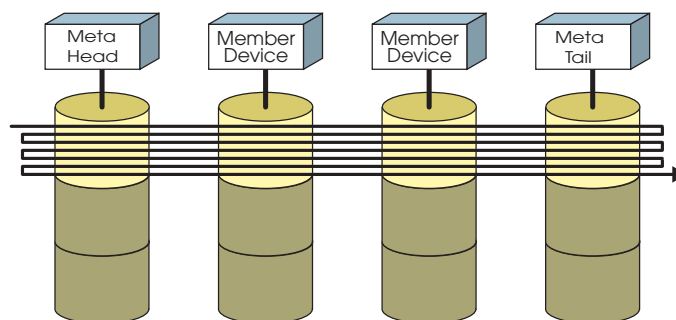


Figure 1-3 Striped Data

Striping data across the multiple drives in definable cylinder stripes benefits *sequential writes* by avoiding stacking multiple writes to a single spindle and disk director. This scheme creates a large volume, but additionally balances the I/O activity between the disk devices and the Symmetrix disk directors.

Consider the following when creating a striped meta device:

Note: This information is important if you plan on creating a new device that will be used as a BCV or the target of a clone operation of a striped meta formed this way.

- ◆ EMC recommends a stripe size of 2 cylinders (which is the requirement for Enginuity Version 5x69 and higher). If you chose a device to be a striped meta member and it has an odd number of cylinders, the last cylinder will not be used.
- ◆ When creating a striped meta device, each member should be the same size. Once a member is full, no further stripes will be written to any member.

Table 1-1 defines the data stripe sizes and capacities for Symmetrix meta volumes.

Table 1-1 Meta Volume Stripe Sizes

Stripe Size	Stripe Capacity (512 Byte Blocks)
2 Cylinders ^a	1,920
4 Cylinders ^b	3,840
8 Cylinders ^b	7,680
16 Cylinders ^b	15,360
32 Cylinders ^b	30,720
64 Cylinders ^b	61,440

a. The recommended default stripe size is 2 cylinders for best performance. Enginuity Version 5669 requires a stripe size of 2 cylinders.

b. Available with Enginuity Version 5x68 or earlier.

Virtual Devices

Snapshot copies use *virtual devices* to support TimeFinder®/Snap operations. A *virtual device* is a host-accessible device containing track-level location information (pointers), which 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.

Note: For more information about virtual devices and TimeFinder/Snap operations, refer to the *EMC Solutions Enabler TimeFinder Family 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 select a particular pool.

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

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

Note: For more information about SAVE devices and TimeFinder/Snap operations, refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide*.

Devices and System Configurations

The SYMCLI maintains configuration and status information for every device on each of the Symmetrix arrays that are accessible from the host.

A Symmetrix device can map to a part of a physical disk or to an entire disk. The part of a physical disk to which a Symmetrix device maps is called a *hyper volume* or a *hyper*.

Configurations

There are various Symmetrix device configurations that involve various mirror set protection possibilities.

Table 1-2 lists the most typical mirror set protection schemes.

Table 1-2 Mirror Set Protection Schemes

Device/Mirror Configuration	M1	M2	M3	M4
2-Way-BCV-Mir	Data	Data		
2-Way-Mir	Data	Data		
3-Way-Mir	Data	Data	Data	
4-Way-Mir	Data	Data	Data	Data
BCV	Data			
DRV	Data			
Parity RAID 3+1 (in multiples of 3)	RAID Data	RAID Parity (shared)		
Parity RAID 7+1 (in multiples of 7)	RAID Data	RAID Parity (shared)		
RAID-5 ^a	Data and Parity	Data and Parity		
RAID-5-BCV ^a	Data and Parity	Data and Parity		
RDF1	Data	Remote Data		
RDF1+R-5 ^a	Data and Parity	Data and Parity		
RDF1-BCV	Data	Remote Data		
RDF1-BCV+R-5 ^a	Data and Parity	Data and Parity		
RDF1-BCV-Mir	Data	Remote Data	Data	
RDF1-Mir	Data	Remote Data	Data	

Table 1-2 Mirror Set Protection Schemes (continued)

Device/Mirror Configuration	M1	M2	M3	M4
RDF1-R-S	RAID Data	Remote Data	RAID Parity	
RDF2	Remote Data	Data		
RDF2+R-5 ^a	Data and Parity	Data and Parity		
RDF2-BCV	Remote Data	Data		
RDF2-BCV+R-5 ^a	Data and Parity	Data and Parity		
RDF2-BCV-Mir-	Remote Data	Data	Data	
RDF2-Mir	Remote Data	Data	Data	
RDF2-R-S	Remote Data	RAID Data	RAID Parity	
Unprotected	Data			

- a. This table is maintained for historical reasons. RAID-5 devices are built on either four or eight hyper members, where data and parity are striped across all members.

Figure 1-4 illustrates the devices for a two-way mirror concatenated meta device configuration (Meta 2-Way-Mir). The meta device is comprised of four mirrored devices of 1 GB each.

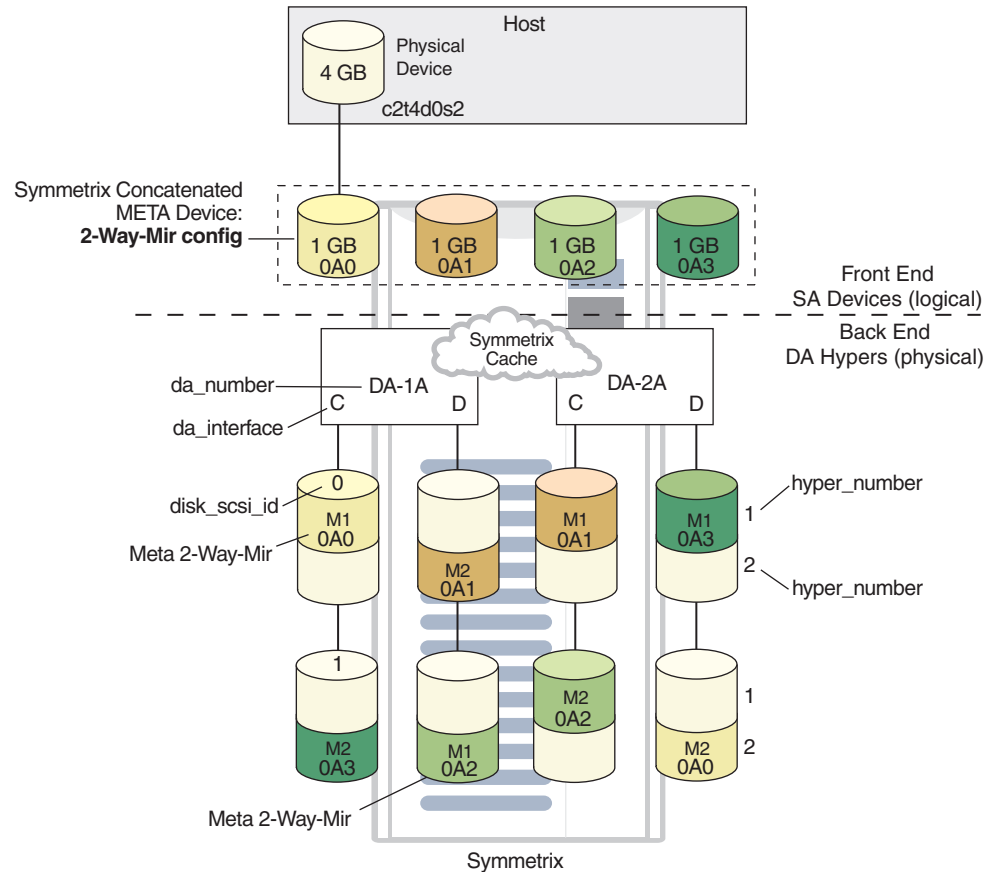


Figure 1-4 Symmetrix Concatenated Meta Device Two-Way Mirror Scheme

Figure 1-5 illustrates the devices for two three-way mirror configurations (3-Way-Mir).

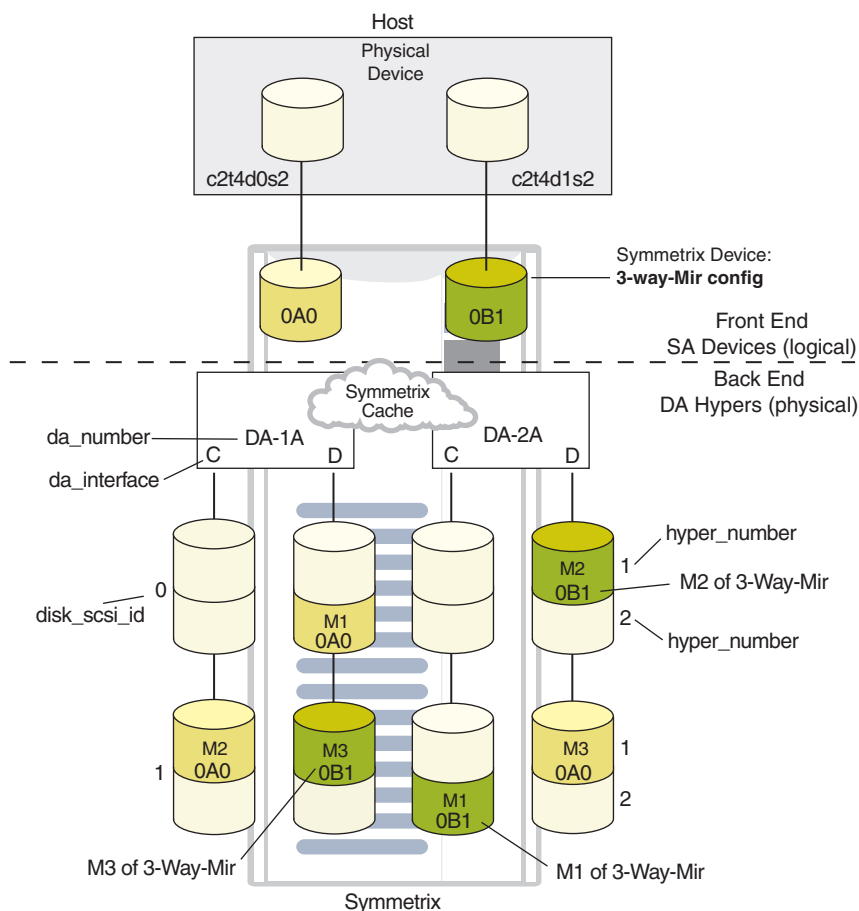


Figure 1-5 Symmetrix Three-Way Mirror Scheme

Parity RAID

Parity RAID devices must be created in multiples of three or seven. For example, a Parity RAID 3+1 protection group requires three devices for data and one for shared parity. Any requests to create parity devices outside of the three or seven multiple factor will be denied. The Symmetrix array maps each of device's data mirrors to the same mirror positions on a separate hyper and creates one parity device to be shared by the data devices. The number of Parity RAID (three or seven members) devices must be defined in the Symmetrix Attributes.

Figure 1-6 illustrates the devices of a parity RAID mirror configuration (3+1).

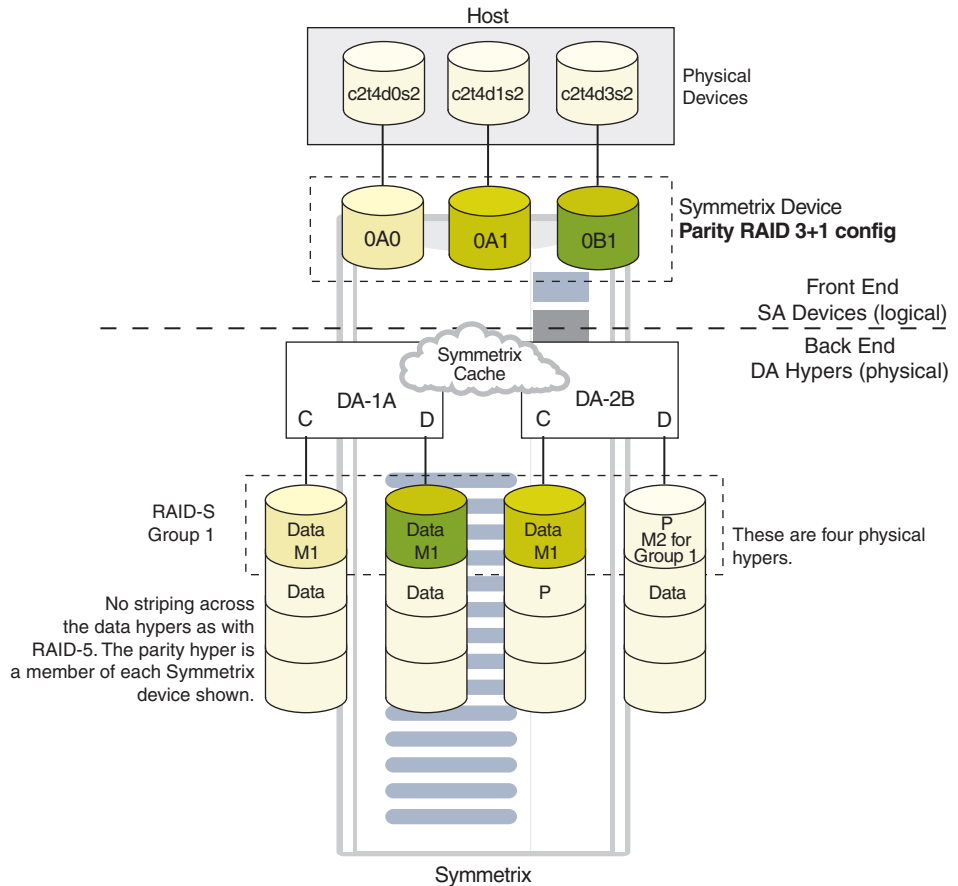


Figure 1-6 Symmetrix Parity RAID Mirror Scheme

You can use the `symdev list` command, to get a list of all the available devices. You can then choose a device from this list and get the configuration scheme it is serving by using the `symdev show` command. This information then leads you to back-end information for the device's disk director(s) and corresponding hyper volumes, and their mapping to disk drives. For more information, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*.

SRDF and BCV Configurations

SRDF Devices

The Symmetrix Remote Data Facility (SRDF) is a business continuance solution that maintains a mirror image of data at the device level in Symmetrix systems located in physically separate sites.

SRDF configurations provide for either a unidirectional or a bidirectional data transfer over SRDF links between sites. Devices are designated as either R1 (source) or a R2 (target) devices to synchronize and coordinate SRDF transfer activity. As shown in Figure 1-7, a source R1 device is designated an RDF1 type, while a target R2 device is designated an RDF2 type. An R1 or an R2 device can also have associated mirror(s) (M1, M2) for increased protection. These are designated RDF1+Mir or RDF2+Mir as configuration types when using `symconfigure`.

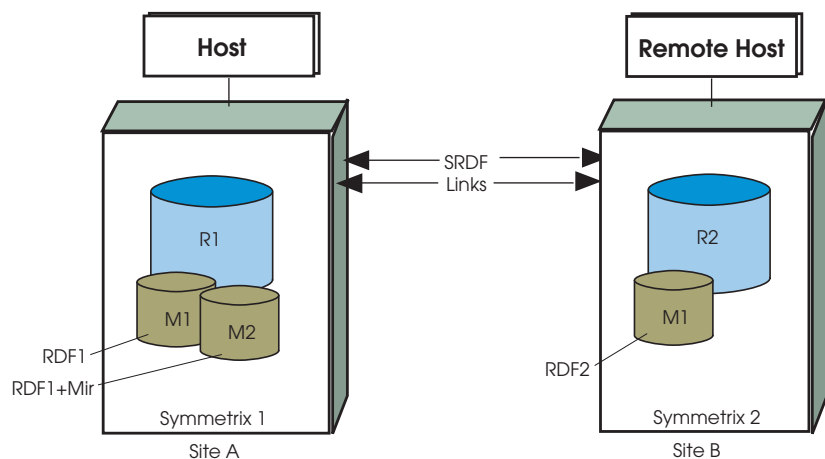


Figure 1-7 Simple SRDF Device Configuration

SRDF and BCV Devices

You can configure both SRDF and TimeFinder (BCV) devices that might be in various compounded remote configurations.

As shown in Figure 1-8 on page 1-16, 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 can also have a third site (remote Site C) to remotely mirror just the BCV devices in the Symmetrix at Site A.

There are two types of SRDF-connected BCV devices:

- ◆ The SRDF-connected BCVs can be paired with R2 mirrors of the local RDF standard devices. In this case, the BCV is designated RBCV, which identifies it as Remote BCV device.
- ◆ The SRDF-connected BCVs can be paired with the R2 mirrors of the local RDF BCV devices, designated RDF1-BCV, also can be designated R1-BCV. The SRDF-connected BCV devices can be remotely associated with one device group of any type, in this case a BCV device designated BRBCV, which identifies this device as a BCV on a remote host originating from a BCV.

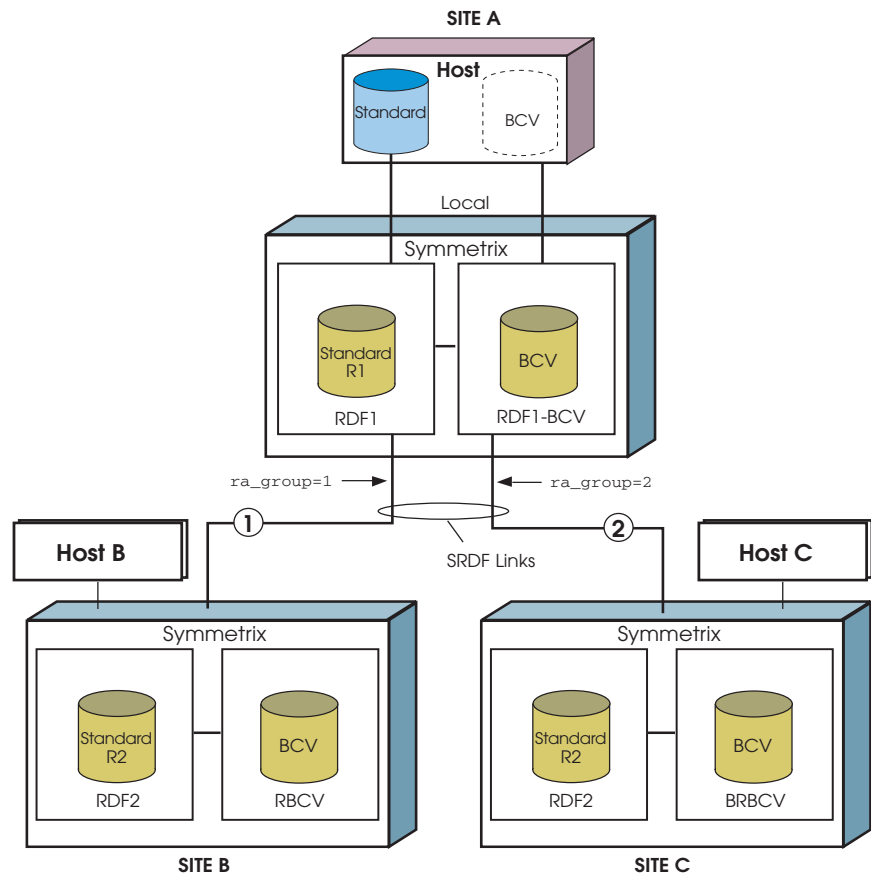


Figure 1-8 Various SRDF-Connected Device Configurations with BCVs

Command Summary

Table 1-3 summarizes the configuration change (`symconfigure`) command actions. For more detail about the command syntactical form, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Table 1-3 Symconfigure Commands Summary

Command	Description
<code>preview</code>	Verifies the syntax and correctness of each individual change defined, and then terminates the session without change execution.
<code>prepare</code>	Performs the preview checks and also verifies the appropriateness of the resulting configuration definition against the current state of the Symmetrix array; the argument then terminates the session without change execution.
<code>commit</code>	Completes all stages and executes the changes in the specified Symmetrix array.
<code>query</code>	Allows the monitoring of progress of a configuration request.
<code>abort</code>	Allows for the termination of a change session left dangling due to the host connection to the configuration server becoming broken.
<code>verify</code>	Verifies that the configuration currently running in the specified Symmetrix array complies with the requirements for the host-based configuration changes.
<code>list</code>	Shows the free physical disk space within the Symmetrix array.

This chapter describes how to perform powerful configuration change operations using SYMCLI.

◆ Configuration Control Operations Overview	2-2
◆ Device Creation Strategy	2-8
◆ Creating Devices	2-13
◆ Deleting Devices.....	2-17
◆ Converting Devices.....	2-19
◆ Setting Device Characteristics	2-22
◆ Mapping Devices	2-24
◆ Unmapping Devices	2-26
◆ Configuring Dynamic RDF-Capable Devices	2-28
◆ Setting RDF Group Attributes.....	2-31
◆ Configuring Meta Devices	2-32
◆ Managing CKD Devices.....	2-39
◆ Managing SAVE Device Pools.....	2-42
◆ Setting Port Characteristics.....	2-45
◆ Setting Symmetrix Metrics.....	2-50
◆ Swapping RA Groups	2-52
◆ Managing Spare Disks.....	2-54
◆ Swapping R1 Devices with R2 Devices	2-56
◆ Stages of the Configuration Change Action.....	2-58
◆ Monitoring the Status of the Change Session	2-61
◆ Aborting the Change Session	2-62
◆ Subsystem IDs for Mainframe Systems	2-63

Configuration Control Operations Overview

The `symconfigure` command and the `symrdf swap` command allow the *advanced* user more powerful control to manage various types of configuration changes affecting Symmetrix devices than otherwise provided by the SYMCLI set.

Most SYMCLI commands, including the `symconfigure` command can operate on Symmetrix systems and devices that are either directly attached to the user's host, or are accessible via an RDF link.



CAUTION

Before issuing a `symconfigure` command to a Symmetrix containing both open system and mainframe devices, you must verify that the mainframe Missing Interrupt Handler (MIH) period for *each* mainframe server attached to the Symmetrix is set to at least 2 minutes.

To view the current MIH period, use the z/OS command `D IOS,MIH`, and note the value for the DASD device class (for example, DASD=1:30).

To change the MIH period, use the z/OS command `SETIOS MIH,DASD=mm:ss`, where `mm:ss` is a period of time in minutes and seconds. Then use the z/OS command `D IOS,MIH` to verify your changes.

Once you have completed your configuration change session, use the `SETIOS MIH,DASD=mm:ss` command to set the MIH period back to its original value.

For more information, refer to your mainframe system administrator.

Device Locking

Symmetrix device locks are used to prevent other control operations (that is, SRDF, TimeFinder) from performing on devices being modified by a configuration change session. Device locks are engaged by the SYMAPI during the *preview* stage. For information on the various phases of running a configuration change session, refer to *Stages of the Configuration Change Action* on page 2-58.

Device locks are retained until the configuration change session has completed and changes have been activated on the Symmetrix. To view a list of the locked devices on the Symmetrix, use the `symdev list -lock` command. Device locking is also engaged by the other Symmetrix control operations for blocking devices during control actions.

Held Devices

EMC's TimeFinder/Snap operations allow for devices to be marked as *held*, to be available for snap session copies. Devices are attached and detached as snap target devices using the `symsnap attach` or `detach` command. Held devices are set as Not Ready to the host and are not available for configuration changes. To view a list of devices currently held, use the `symdev list -held` command.

Operations Classes

The following are classes of configuration control operations available when applying the `symconfigure` command:

Note: For information on the Engenuity requirements for the configuration control operations, refer to the *EMC Solutions Enabler Version 6.0 Support Matrix*.

Creating devices

Creates new Symmetrix devices.

Converting a device's configuration

- ◆ Converts a device configuration by adding or removing BCV or DRV attributes.
- ◆ Converts an RDF device from static RDF to dynamic RDF.

Changing a device's protection

Increases or decreases device protection by adding or removing a mirror(s).

Deleting devices

Deletes Symmetrix devices.

Configuring RDF

- ◆ Adds/removes RDF attributes.
- ◆ Swaps RDF source/target attributes for an RA group.
- ◆ Enables/disables SRDF/A on an RA group.
- ◆ Sets RA group attributes.
- ◆ Converts static RDF device to dynamic.

Managing Meta devices

- ◆ Forms/dissolves meta devices.
- ◆ Adds/removes meta members.
- ◆ Converts from concatenated to striped and vice versa.
- ◆ Modifies stripe size.

Managing SAVEDEV pools

- ◆ Adds/deletes pools
- ◆ Adds/removes pool members
- ◆ Enables/disables pool members

Managing Spare disks

- ◆ Adds a new spare disk.
- ◆ Removes an existing spare disk.

Mapping devices to ports

- ◆ Maps/unmaps a Symmetrix device to a front-end port.
- ◆ Maps/unmaps a range of Symmetrix devices that form a mainframe CU image to a front-end port.
- ◆ Copies device mappings from one EA/EF port to another.
- ◆ Add or remove PAV aliases from mainframe devices.

Setting device attributes

Sets the device attributes.

Setting device emulation

Sets the device emulation type.

Setting front-end port attributes

- ◆ Sets/resets SCSI or fibre port flags.
- ◆ Sets fibre FA loop ID.
- ◆ Sets hostnames of port connections.
- ◆ Sets Gig-E front-end port IP addresses and netmasks

Setting the Symmetrix metrics

Sets the metrics of Symmetrix configuration.

Note: The overall processing time for the changes can vary from 5 minutes to over an hour, depending on the class of changes. RDF changes will also be applied to the remote Symmetrix array. Even more time is needed if the devices are to be synchronized.

Configuration Change Safety

Before making configuration changes, you must thoroughly understand your Symmetrix configuration. The following guidelines help to establish safe disciplines as you begin any change that can easily impact stored data:

1. Verify that the current Symmetrix configuration is a viable configuration for host-initiated configuration changes.

To verify your current Symmetrix configuration is ready for changes, enter:

```
symconfigure verify -sid SymmID
```

2. Before creating new Symmetrix devices, check for free physical disk space using the following command:

```
symconfigure list -freespace [-units CYLINDERS|MB] \
-sid SymmID
```

Free disk space on unformatted disks is shown as available for all emulation modes. New Symmetrix devices will be created first on physical disks that have no prior allocations, causing these disks to be committed to that emulation type.

Examine the distribution of freespace across formatted disks to see if the desired mirroring can be provided using one of the following commands:

```
symdev list -sid SymmID -da all -space
or
symdisk list -sid SymmID
```

3. No configuration change is activated in the Symmetrix system until you *commit* the action.

Some classes of change operations may or may not impact current I/O. When possible, before you *commit* any action, stop I/O activity on the Symmetrix devices to be altered during a configuration change session. Consider the *Impact on I/O* statement for each class of operation in this chapter before working on any online devices.

If required, set the impacted device(s) for change to be Not Ready using the following command form:

```
symld -g DgName not_ready [LdevName [LdevName... ]]
```

Note: If I/O activity on an affected Symmetrix device occurs before or during a `commit` action, the `commit` action might fail. At the very least, heavy I/O activity on unaffected devices will impact how long it takes to commit changes.

4. Ensure that all your critical data is preserved and safe when creating new or changing device configurations. Do not store data on any device that is not mirrored.

All final configurations and device attribute adjustments must meet certain open systems guidelines put forth in the tables of the *EMC Symmetrix Support Matrix*. You can find the *EMC Symmetrix Support Matrix* on the EMC Powerlink™ website at:

<http://powerlink.emc.com>

Note: Contact the EMC Customer Support Center for assistance in reverting to your previous configuration should there be unforeseen problems with the new configuration.

5. After committing a `symconfigure` mapping operation, you must update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors. For more detailed information, refer to the *Open Systems Host Environment Product Guide*.

To update your hosts, run the utilities specified for your platform as described in *Updating the Host* on page A-1. The `symcfg discover` command should then be issued to update the SYMAPI database with the new device mapping information. After the host environment is updated, I/O activity can resume with the Symmetrix device.

The Command File

For the configuration change session, you must define a set of changes within a specified command file. The `symconfigure` command file format contains various command entries terminated with a semicolon (;). Multiple changes can be made in one session, but all changes must fall into one operation class. Command file examples are provided with the actions described later in this chapter.

Note: Starting with Enginuity Version 5669, changes for multiple operation classes can be executed within the same session, except for dynamic RDF changes and SAVE device pool changes.

Device Creation Strategy

Creation of certain device (configuration) types may require more than one class of command execution. Table 2-1 describes the classes or steps required to create a device type and/or convert the various types of devices from one configuration type to another.

Table 2-1 Steps to Create a Device

Desired Device Type	Class - Step 1 create	Class - Step 2 convert
2-Way-BCV-Mir	2-Way-BCV-Mir	
2-Way-Mir	2-Way-Mir	
3-Way-Mir	3-Way-Mir	
4-Way-Mir ^a	4-Way-Mir	
BCV	BCV	
CKD-Meta	CKD-Meta (in multiples of 4)	
DRV	DRV	
Parity RAID	RAID-S (in multiples of 3 or 7)	
RAID-5	RAID-5	
RAID-5-BCV	RAID-5-BCV	
RDF1	RDF1	
RDF1-Mir	RDF1-Mir	
RDF1+R-5	RDF1+R-5	
RDF1-BCV	BCV	→ RDF1-BCV
RDF1-BCV+R-5	RDF1+R-5	→ RDF1-BCV+R-5
RDF1-BCV-Mir	RDF1+Mir	→ RDF1-BCV-Mir
RDF2	RDF2	
RDF2-Mir	RDF2-Mir	
RDF2+R-5	RDF2+R-5	
RDF2-BCV	BCV	→ RDF2-BCV

Table 2-1 Steps to Create a Device (continued)

Desired Device Type	Class - Step 1 create	Class - Step 2 convert
RDF2-BCV+R-5	RDF2+R-5	→ RDF2-BCV+R-5
RDF2-BCV-Mir	RDF2+Mir	→ RDF2-BCV-Mir
Unprotected	Unprotected	
Virtual ^b	VDEV	

a. 4-Way-Mir devices cannot have any BCV devices.

b. Requires that you contact EMC.

For information on creating dynamic-RDF capable devices, refer to *Configuring Dynamic RDF-Capable Devices* on page 2-28.

When you create an RDF device where a corresponding device must be created on the remote SRDF-linked Symmetrix, a configuration change session is initiated and managed on the remote Symmetrix by the local `symconfigure` utility.

Note: SAVE devices are created following the same procedure as for standard protected devices, with the additional specification of `attribute=savedev`.

Device Creation Restrictions

You can create devices with a variety of emulation types: Fixed Block Architecture (FBA), Celerra® FBA or VME512 FBA, and CKD for z/OS environments.

You cannot create or convert to SRDF device types:

- ◆ When Domino mode is enabled on any current SRDF pairs.
- ◆ When there are any invalid tracks on any of the current SRDF devices.
- ◆ When concurrent RDF is enabled on a device.

When creating devices in an z/OS environment, all new devices must have the same MVS SSID and any remote devices must also have the same remote MVS SSID within a single session. To create devices with different SSID values, you will need to run more than one session.

The creation of meta devices often requires more than one configuration change session. Some require three sessions (*create*, *form*, and *convert*) as shown in Table 2-2.

Table 2-2 Steps to Create a Meta Device from One Type to Another

Desired Device Type	Session 1 <i>create</i>	Session 2 <i>form</i>	Session 3 <i>convert</i>
Meta 2-Way-Mir	2-Way-Mir	→ Meta 2-Way-Mir	
Meta 3-Way-Mir	3-Way-Mir	→ Meta 3-Way-Mir	
Meta 4-Way-Mir	4-Way-Mir	→ Meta 4-Way-Mir	
Meta Parity RAID	RAID-S (in multiples of 3 or 7)	→ Meta Parity RAID	
Meta RDF1	Unprotected	→ Meta Unprotected	→ Meta RDF1
Meta RDF2	Unprotected	→ Meta Unprotected	→ Meta RDF2
Meta RDF1+R-S	RAID-S (in multiples of 3 or 7)	→ Meta Parity RAID	→ Meta RDF1+R-S
Meta RDF2+R-S	RAID-S (in multiples of 3 or 7)	→ Meta Parity RAID	→ Meta RDF2+R-S
Meta RAID-5	RAID-5	→ Meta RAID-5	
Meta RAID-5 BCV	RAID-5 BCV	→ Meta RAID-5 BCV	
Meta BCV	BCV	→ Meta BCV	
Meta 2-Way-BCV-Mir	2-Way-BCV-Mir	→ Meta 2-Way-BCV-Mir	
Meta RDF1-BCV	BCV	→ Meta BCV	→ Meta RDF1-BCV
Meta RDF2-BCV	BCV	→ Meta BCV	→ Meta RDF2-BCV
Meta 2-Way-Mir-RDF	2-Way-Mir	→ Meta 2-Way-Mir	→ Meta 2-Way-Mir-RDF
Meta VDEV	VDEV	→ Meta VDEV	

Meta Configuration Restrictions

All meta volumes (devices) must be unmapped before configuring them as either a meta head or a member. If not, the `form` command for meta devices will be rejected.

Note that RDF devices cannot be used to form meta devices in one step. When this is necessary, the RDF devices must first be converted to non-RDF devices (as shown Table 2-2 on page 2-10), the meta devices must be formed separately on both Symmetrix arrays, and then the meta head on one Symmetrix array can be converted back to an RDF device.

Also, once the RDF meta device has been established, it cannot be modified in one step. If any modifications are required, it will again have to be converted to a non-RDF device first.

Virtual Device Configuration Restrictions

Symmetrix virtual devices are configured as full size devices, but use minimal disk storage on the back end. Virtual devices must be mirrored devices and cannot be grouped with SAVE devices. The format must be either CKD or FBA. Virtual devices can also be used to form a VDEV meta device. Other Symmetrix devices cannot be converted to form virtual devices. To configure a virtual device, set `config=vdev` in the device file.

Note: You cannot create VDEV devices. For more information, contact your EMC Representative.

SAVE Device Configuration Restrictions

Symmetrix SAVE devices work in conjunction with virtual devices and can be unprotected¹, mirrored, or RAID-5.² SAVE devices cannot be part of a meta volume or grouped. Other Symmetrix devices cannot be converted to form SAVE devices. Configuring virtual and SAVE devices requires careful planning to anticipate what percentage of the data may change during a virtual copy session. Creating the SAVE device pool too small could result in data changes being lost during a copy session. Make sure that you create enough SAVE devices and set the size of the SAVE devices large enough to handle

-
1. Configuring SAVE devices as unprotected is not recommended and requires special approval by EMC.
 2. For a given device emulation type, all SAVE devices must have the same protection.

any expected changes. To configure a SAVE device, set `attribute=savedev` in the request to create a new device.

RDF Configuration Changes

Making changes to RDF device configurations and swapping RA groups requires that a corresponding change be made on the remote Symmetrix array. The `symconfigure` command performs these changes in parallel. If another user is executing a configuration change on the remote Symmetrix, the RDF change will not be applied to either side.

Changes to Mirror Protection

The mirroring protection of a device can be increased using a separate session from BCV, DRV, and RDF change sessions. Device configuration changes to apply or remove the BCV, DRV, or RDF attribute that results in a change to the device's mirroring, will be denied.

For example, converting a 2-Way-BCV to an unprotected device will be denied. This involves two changes being made to a device, removing a mirror and removing the BCV attribute. For this type of configuration, you can only remove a mirror to form a BCV device or remove the BCV attribute to form a 2-Way-MIR device.

The mirroring protection of a device can also be decreased, and this too must be done via a separate session from the BCV, DRV, and RDF change sessions.

As a result of removing mirrors from a device, a new device is created from the discarded mirrors, resulting in a new Symmetrix device. If the original or new device is unprotected, it cannot be mapped to a host.

Creating Devices

In the command file, you can create (add) one or more Symmetrix devices in a specified Symmetrix array.

To create one or more devices, use the following command:

```
create dev count=n, size=Cylinders,
        emulation=EmulationType,
        config=DeviceConfig,
        [, remote_config=DeviceConfig, ra_group=n,
          [remote_mvs_ssid=nnn] ]
        [, mvs_ssid=nnn]
        [, attribute=ckd_meta | savedev
          [in pool PoolName] [member_state=ENABLE | DISABLE]
        [, disk_group_num=nnn, remote_disk_group_num=nnn];
```

where:

count = the number of devices to create.

size = the size of the device needed in number of cylinders. A cylinder of an FBA device contains 960 512-byte blocks or, 49 megabytes.

To calculate the number of cylinders, use either of the following:

```
blocks ÷ 960
or
(devicesize in megabytes) × 10242 ÷ (960 × 512 bytes)
or
(size in megabytes) × 2.133
```

Note: Devices to be used as BCV, RDE, or meta members will need to precisely match corresponding device sizes. Use `symdev/sympd show` or `symdev/sympd list -cyl` commands to see relevant device sizes.

For additional information about sizing devices in cylinders, refer to Chapter 1 of the *EMC Solutions Enabler Open Storage Base Management CLI Product Guide*.

emulation = the device emulation type, which currently must be Fixed Block Architecture (FBA), CELERRA FBA, VME512 FBA or CKD for z/OS environments. For a list of supported AS/400 environments, refer to the *symconfigure* description in the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

config = the desired device configuration type. Possible values are shown in the second column of Table 2-1 on page 2-8.

remote_config = the desired remote SRDF configuration (if any) from Table 2-1 on page 2-8.

ra_group = the RA group number in the SRDF environment.

remote_mvs_ssid = When creating an RDF device in a remote Symmetrix array that also contains CKD devices, a z/OS (MVS) subsystem ID (*remote_mvs_ssid*) value may be provided so the new FBA devices are not seen as part of an existing subsystem ID group.

Only one *mvs_ssid* and *remote_mvs_ssid* can be used in a session. They will be applied to all devices created within that session.

When using the `convert device` command to reduce mirroring, the removed mirror becomes a new stand-alone Symmetrix device. An attempt will be made to use the same MVS SSID for the new device. If the MVS SSID group is full, you must supply a new MVS SSID.

When mapping a set of devices to an EA or EF port, the current *mvs_ssid* assigned to the devices may need to be changed. If the devices are becoming part of an existing CU image, they will be assigned the *mvs_ssid* of the devices already mapped. If a new CU image is being formed and mapped, a new *mvs_ssid* can be assigned during the map request. It is not valid to have some devices with a particular SSID mapped and some unmapped.

mvs_ssid = When creating a device in a Symmetrix array that also contains CKD devices, a z/OS (MVS) subsystem ID (*mvs_ssid*) value can be provided so the new FBA devices are not seen as part of an existing subsystem ID group. For more information about SSIDs, refer to *Subsystem IDs for Mainframe Systems* on page 2-63.

To obtain a list of SSIDs currently in use, use the following:

```
symcfg -ssid list
```

ckd_meta = When creating a device with emulation type of CKD-3380 or CKD-3390, indicates that the device should be a striped meta device. CKD meta devices must be created in sets of four devices.

savedev = When creating a device, this indicates that the device should be a SAVE device. The device will become part of a pool of devices that are used with TimeFinder/Snap for virtual device Snap operations.

PoolName = The name of the SAVE device pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-), and underscore (_) characters. The name `DEFAULT_POOL` is reserved to represent the container of all *unpooled* devices.

disk_group_num = When creating a device, this option allows you to specify a disk group. A disk group is a set of physical disks set aside to be used to create devices of the protection level. Disk group usage may improve the performance of some configurations.

To see how disk groups are organized in your configuration, use the following commands:

```
symdisk -sid SymmID list -by_diskgroup
and
symdev -sid SymmID list -diskgroup(nn)
```

remote_disk_group_num = When creating a device, this option allows you to specify a remote disk group.

Example: Creating Devices

To add four new Symmetrix devices to Symmetrix array 000000012345 as two-way mirrored devices with a size of 1100 cylinders (516 MB), using FBA emulation, and to create the hypervisors to support the new devices in disk group 3, create a command file containing the following option:

```
create dev count=4, size=1100,emulation=FBA,
config=2-Way-Mir disk_group=3;
```

Then activate the option using the command:

```
symconfigure -sid SymmID -file create, cmd -v -hop
commit
```

Impact on I/O

When creating devices, there is no impact on any I/O activity.

Enable SYMAPI Environment Option

Environment option `SYMAPI_CTRL_OF_NONVISIBLE_DEVS` in the `options` file must be enabled. If this option is disabled, the created devices, which are nonvisible, will be blocked from having changes performed on them. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

**Restrictions on
Creating Devices**

There are restrictions on the use of unprotected devices created by the `symconfigure` utility, or converted to unprotected devices. Such devices cannot be mapped to hosts for normal data storage use. If an unprotected device is intended for use as a gatekeeper, it should be created with 48 cylinders or less in order to be mapped to a host.

Deleting Devices

In the command file, you can delete one or more Symmetrix devices from the specified Symmetrix array. Deleting a device frees the space that it once occupied for future use.

To delete one or more devices, use the following command:

```
delete dev SymDevName[:SymDevName]
[,raidset = TRUE];
```

where:

raidset = When requesting to delete a RAID-S group, set **raidset** to **TRUE** and list the first RAID-S member. It is not necessary to list the other members.

Example: Deleting Devices

To delete Symmetrix device 0015 from Symmetrix array 000000012345, create a command file containing the following:

```
delete dev 0015;
```

Then activate the option using the command:

```
symconfigure -sid 0015 -file delete, cmd -v -hop commit
```

Example: Deleting a RAID-S Set

To delete all devices in a RAID-S set, create a command file with a statement that only identifies the first member of the RAID-S set:

```
delete dev 015C, raidset = TRUE;
```

Then run the **symconfigure** utility:

```
symconfigure -sid 05605 -file delete_raids.cmd -v -hop
commit
```

All members will be deleted.

Restrictions on Deleting Devices

The following are restrictions for deleting devices:

- ◆ The Symmetrix array must be running Engenuity Version 5670 or higher.
- ◆ The device emulation must be one of the following types:
 - SYMAPI_EMULATION_FBA
 - SYMAPI_EMULATION_CELERRA_FBA
 - SYMAPI_EMULATION_VME_512_FBA
 - SYMAPI_EMULATION_CKD_3380
 - SYMAPI_EMULATION_CKD_3390

- SYMAPI_EMULATION_AS400_590
- SYMAPI_EMULATION_AS400_590R
- SYMAPI_EMULATION_AS400_6713_30
- SYMAPI_EMULATION_AS400_6713_50
- SYMAPI_EMULATION_AS400_6717_50
- SYMAPI_EMULATION_AS400_9337_5AA
- SYMAPI_EMULATION_AS400_9337_5AC
- ◆ The device must not have an attached BCV or DRV.
- ◆ The device must not have any snap sessions.
- ◆ The device must not be:
 - Already deleted
 - Mapped to a front-end port
 - Part of an RDF consistency group
 - A source or target of a clone session
 - Held
 - A virtual device that is in use
 - An RDF device
 - A meta member
 - An SFS device
 - A SAVE device¹
 - WORM-protected
 - The VCM database device
 - Masked by VCM
- ◆ You cannot delete one member of a raidset without deleting all the members.

1. Starting with Solutions Enabler Version 6.0, you can delete a SAVE device as long as it is disabled with no active tracks.

Converting Devices

In the command file, you can convert the configuration type of an existing device or devices using the following command:

```
convert dev SymDevName[:SymDevName] to DeviceConfig
[ ra_group=n, remote_dev=SymDevName,
  invalidate=R1|R2|NONE, start_copy=YES|NO ],
[mvs_ssaid=nnn] [raidset = TRUE];
```

To convert an RDF device or devices from static RDF to dynamic RDF, use the following command:

```
convert rdf dev SymDevName[:SymDevName] to dynamic;
```

where:

SymDevName = the Symmetrix device name of the device targeted for change. To target more than one device, indicate the first and last devices in a series separated by a colon (:).

DeviceConfig = the desired device configuration type. Possible values are shown the first column of Table 2-1 on page 2-8.

ra_group = the RA group number in the SRDF environment.

remote_dev = the remote Symmetrix device name of the particular device targeted for change. If you specify a range of *SymDevNames* in the first line of the convert statement, the remote *SymDevName* value will be increased incrementally to arrive at the corresponding device number.

invalidate = the RDF device (R1 source or R2 target) to invalidate. An invalidated copy requires a full refresh (synchronization of all tracks) at some point. If *NONE* is selected, the RDF device must be specified later using the *symrdf* command.

start_copy = whether an RDF pair should be synchronized after the configuration change is committed.

Note: When creating RDF devices, all conversions within a session must have:

- DeviceConfig settings that reflect the same destination RDF type (RDF1 or RDF2).
 - The same *ra_group* number.
 - The same *invalidate* option.
 - The same *start_copy* option.
-

mvs_ssid = Specifies the z/OS (MVS) subsystem ID that will be assigned to any device created as a result of removing any mirror(s). If not provided, the original MVS SSID will be assigned when available.

raidset = When requesting to convert a RAID-S group to unprotected devices, set **raidset** equal to **TRUE** and list the first RAID-S member. It is not necessary to list the other members.

Example: Converting BCV Devices

To convert two existing BCV devices (001C and 001D) to an RDF1-BCV configuration and to invalidate the source R1 and synchronize the RDF pair, enter in the command file:

```
convert dev 001C:001D to RDF1-BCV, ra group=1,
remote_dev=001c, invalidate=R1, start_copy=YES;
```

Example: Converting RDF Devices

To convert RDF device 0014 from static RDF to dynamic RDF, enter in the command file:

```
convert rdf dev 0014 to dynamic;
```

Example: Converting a RAID-S Set to Unprotected

To convert a RAID-S set whose members are RDF devices (either RDF1+R-S or RDF2+R-S) to unprotected RDF devices, create a command file that requests any member of the RAID-S be converted:

```
convert dev 00D5 to RDF1, raidset=TRUE;
```

Then run the **symconfigure** utility:

```
symconfigure -sid 05605 -file cvt_raids.cmd -v -nop
commit
```

All members will be converted.

Note: If there is a problem converting any member of a RAID set, the SYMAPI log file will contain the details.

Impact on I/O

The following are restrictions/conditions to avoid impact on I/O activity:

- ◆ When adding DRV attributes, devices must be unmapped.
- ◆ When adding/removing RDF attributes, there are no restrictions on I/O. The RDF pair must be split or failed over. If failed over, the R1 device must be unmapped.
- ◆ When adding/removing BCV attributes, there are no restrictions on I/O. The standard/BCV pair must be split.

Device Group Invalid Warning

When converting devices that are currently in a device group, some groups will be declared invalid after particular operations.

One example would be if you are performing TimeFinder operations and have placed in a group, one standard (STD) device and one TimeFinder (BCV) device. `symconfigure` enables you to change the BCV device to a non-BCV device. This conversion would leave the group in an invalid state.

There are two ways to handle this situation:

- ◆ Remove the TimeFinder (BCV) device from the group before you begin the conversion process.
- ◆ Or, after the device has been converted, export (`symdg export`) the device group and remove from the file `symdg.txt` the invalid device and import (`symdg import`) the device group.

Enable SYMAPI Environment Option

Environment option `SYMAPI_CTRL_OF_NONVISIBLE_DEVS` in the `options` file must be enabled (or not present in the options file) if the device is not currently mapped to the user host or you are working on DRV devices. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Restrictions

The following are restrictions for converting devices:

- ◆ The `convert` command can be used for three different classes of device configuration changes, as long as the class types are performed in separate sessions.

Note: Starting with Engenuity Version 5669, changes for multiple operation classes can be executed within the same session, except for dynamic RDF changes and SAVEDEV pool changes.

The three class types that cannot be used in the same session are:

- Add/remove BCV/DRV attributes
- Add/remove RDF attributes
- Increase/decrease mirroring
- ◆ Only the head of a meta device can have its type changed. The meta member(s) will automatically have the changes applied.
- ◆ R1/R2 swap operations are not supported for Engenuity Version 5669 (or higher) configurations where the R2 device is larger than the R1 device.
- ◆ You cannot convert one member of a raidset to unprotected without converting all the members to unprotected.

Setting Device Characteristics

You can set the device attributes or emulation of a number of devices in a range using the following command:

```
set device SymDevName[:SymDevName]
    [emulation=EmulationType]

    [attribute=[NO] device_attr];
```

where:

emulation = the device emulation type, which can be fixed block architecture selections: FBA, CELERRA FBA or VME512 FBA.

attribute = a device attribute that restricts how a device can be accessed. Possible values include:

- RDB_Cksum
- VCMdb (for device masking)
- Worm (can be enabled only)
- SCSI3_persist_reserv (persistent group reservation)
- dyn_rdf (This option provides the most flexibility in performing dynamic RDF operations)
- dyn_rdf1_only*
- dyn_rdf2_only*

* These options allow a device to become an R1 or R2 device only; therefore, using them prevents the ability to perform RDF swaps.

Example: Setting Emulation

To convert five devices (0015 to 0019) to Celerra FBA emulation, enter in the command file:

```
set device 0015:0019
    emulation=CELERRA_FBA;
```

Example: Setting Attribute

To enable five devices (0015 to 0019) for WORM protection, enter the following in the command file:

```
set device 0015:0019
    attribute=Worm;
```

Example: Resetting Attribute

To reset the device attribute of five devices (0015 to 0019) removing RDB_Cksum, enter the following in the command file:

```
set device 0015:0019
    attribute=NO RDB_Cksum;
```

Enable SYMAPI Environment Option

The environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if the device is not mapped to the user host. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

You can set emulation and attributes in the same session (command file). For more information on classes, refer to *Operations Classes* on page 2-3.

Note: If you are running Engenuity Version 5668 or lower, you cannot execute changes for multiple operation classes within the same session.

Note: The `SCSI_persist_reserv` attribute is only used when you have a Sun Cluster 3.0 environment, where two or more channels access the device.

Restrictions

The following are restrictions that exist while setting device characteristics:

- ◆ When setting the device emulation type, the devices must be unmapped. No I/O to the devices involved.
- ◆ When setting the attribute type to RAD, the devices must be unmapped. No I/O to the devices involved.
- ◆ When setting the attribute type to a mapped device, it is recommended that you minimize the I/O activity to the affected devices.

Mapping Devices

You can map devices to front-end director ports. This is part of the Symmetrix Device Reallocation (SDR) feature.

To map a Symmetrix device to a director port, use the following command:

```
map dev SymDevName to dir DirectorNum:PortNum
    [target=ScsiTarget,] lun=ScsiLun
    [, vbus=FibreVbus]
    [, device_number=ckd_device_number]
    [, awwn=awwn|wwn=wwn|iscsi=iscsi];
```

where:

target = the SCSI target ID (hex value).

lun = the SCSI logical unit number (hex value).

vbus = the virtual bus (vbus) address for mapping to an FA port if using volume set addressing.

device_number = the CKD device number, when mapping a CKD device to an OS/390 host.

awwn = the user-given name or alias WWN of a host HBA port, if updating a VCM database.

wwn = the unique 64-bit World Wide Name (WWN) identifier for an HBA port, if updating a VCM database.

iscsi = the iSCSI name, if updating a VCM database.

Example: Mapping Class

To map meta head (device) 0030 to director 16A, port 0, and SCSI target/lun 0,7, enter in the command file:

```
map dev 0030 to dir 16A:0
    target=0, lun=7;
```

To obtain a list of used addresses, including the next available address, use the following command:

```
symcfg list -SA all -address -available
```

To map a device 0032 to director 16A, port 0, and SCSI target/lun 0,2 and update the device masking database by specifying the WWN 20000000c920b484 of the host bus adapter (HBA) port through which a host accesses the device, enter the following command:

```
map dev 0032 to dir 16A:0 target=0, lun=2,  
    wwn=20000000c920b484;
```

Impact on I/O

The following are restrictions/conditions to avoid impact on I/O activity:

- ◆ When mapping, there are no restrictions on I/O if adding a second path.
- ◆ After committing a `symconfigure` mapping operation, you must update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors. For more information, refer to the *EMC Solutions Enabler Open Storage SRM CLI Product Guide*.
- ◆ To update your hosts, run the utilities specified for your platform as described in *Updating the Host* on page A-1. After the host environment is updated, I/O activity can resume with the Symmetrix device.

Enable SYMAPI Environment Option

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if the device is not mapped to the user host. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Unmapping Devices

You can unmap devices from front-end directors ports. This is part of the Symmetrix Device Reallocation (SDR) feature.

To unmap devices from a director port, use the following form:

```
unmap dev SymDevName[:SymDevName] from dir
<ALL:ALL | ALL:PortNum |
DirectorNum:ALL | DirectorNum:PortNum>
[ devmask_access=<remove|retain>];
```

where:

devmask_access = indicates whether the device masking database should be updated. The `remove` option indicates that device masking access entries for the device should be removed from the VCMDB. The `retain` option specifies that entries remain in the database.

Example: Unmapping Class

To unmap devices 0030 and 0031 from director 16A, port 0, and remove the entries from the device masking database, enter in the command file:

```
unmap dev 0030:0031 from dir 16A:0
devmask_access=remove;
```

Impact on I/O

The following are restrictions/conditions to avoid impact on I/O activity:

- ◆ When unmapping, no I/O activity is allowed on any devices in the specified mapped path. Devices must be made Not Ready or Write Disabled.

For example, to make the device Not Ready:

```
symdmg create -type [REGULAR | RDF1 | RDF2] DgName
symld -g DgName -sid SymmID add dev SymDevName
symld -g DgName not_ready
```

If you are unmapping only one path to a multipathed device, you may prefer to write disable that path only:

```
symld -g DgName -SA 16A -p 0 write_disable
```

It is possible to make a device Not Ready without creating a device group:

```
symdev -sid SymmID not_ready SymDevName
```

Note: Do not attempt to use the `write_disable` argument with the `symrdf` command, as this write disables the source (R1) device(s) or the target (R2) device(s) to its/their local hosts.

- ◆ After committing a `symconfigure` mapping operation, you must update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors. For more information, refer to the *EMC Solutions Enabler Open Storage SRM CLI Product Guide*.

To update your hosts, run the utilities specified for your platform as described in *Updating the Host* on page A-1. After the host environment is updated, I/O activity can resume with the Symmetrix device.

Configuring Dynamic RDF-Capable Devices

With Enginuity Version 5568 and higher, devices can be configured via `symconfigure` to be Dynamic RDF-capable devices. Dynamic RDF functionality enables you to create, delete, and swap SRDF pairs while the Symmetrix array is in operation. Using Dynamic RDF technology, you can establish SRDF device pairs from non-SRDF devices, and then synchronize and manage them in the same way as configured SRDF pairs.

The *Dynamic RDF* feature must be enabled in the Symmetrix array. If using Enginuity Version 5568, configuration options can be done using the `symconfigure` command to mark standard devices as being capable of supporting dynamic RDF operations.

In addition, when enabled, you can dynamically swap the RDF device designations of a specified device group using the `symrdf` command. Source R1 device(s) become target R2 device(s) and target R2 device(s) become source R1 device(s). Swaps using Dynamic RDF perform faster.

If using Enginuity Version 5567, dynamic RDF operations are limited to swap only and the Symmetrix configuration can be changed only by EMC.

Note: Dynamic swap is not supported if Enginuity Versions 5567 and 5568 are mixed between your local and remote Symmetrix.

For information on using the devices with RDF commands, refer to the *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.

There are two steps that must be performed before dynamic RDF operations can be performed:

1. The Symmetrix must have the Dynamic RDF feature enabled.
2. The devices on which you want to perform Dynamic RDF operations must have the Dynamic RDF attribute set.

These actions must be performed on both the local and remote Symmetrix arrays.

Enabling Dynamic RDF on the Symmetrix

To enable Dynamic RDF in a Symmetrix, create a command file containing the following option:

```
set symmetrix dynamic_rdf = enable;
```

Then activate the option using the following format:

```
symconfigure -sid SymmID -file set_sym_dyn_rdf.cmd  
-nop -v commit
```

The current setting of this metric can be viewed using the following format:

```
symconfigure -sid SymmID list -v
```

Sample output:

```
Symmetrix ID : 000187900039

Configuration Server Version : 5670.58.56
Configuration Server Protocol : 0x406
Configuration Server Date : 11.04.2003

FBA formatted : 16683874
FBA unformatted : 0
AS/400 formatted : 0
AS/400 unformatted : 0
IBM_FBA formatted : 0
IBM_FBA unformatted : 0
UNISYS_FBA formatted : 0
UNISYS_FBA unformatted : 0
ICL formatted : 0
ICL unformatted : 0
CKD-3380 formatted : 0
CKD-3380 unformatted : 0
CKD-3390 formatted : 0
CKD-3390 unformatted : 0

Max Hypers per Disk : 67
FBA Multi Access Cache : Enabled
Dynamic RDF Configuration : Enabled
RAID-S support : Enabled
RAID membership count : 3
VCmdb Access Restricted : Disabled
RAID-5 support : Enabled
```

Enabling Dynamic RDF on the Device

Next, identify the non-RDF devices that you want available for use as dynamic RDF devices. Create a command file that enables the dynamic RDF operations for those devices. The following possible attributes are:

```
dyn_rdf          (Using the dyn_rdf option provides the most flexibility)
dyn_rdf1_only   *
dyn_rdf2_only   *
```

* These options limit a device to either an R1 or R2 device, and therefore, using them prevents the ability to perform RDF swaps.

The following example shows such a file:

```
set dev 0020:002A attribute=dyn_rdf;
set dev 0040:0045 attribute=dyn_rdf;
set dev 0056:005A attribute=dyn_rdf;
```

Then, activate the option using the following command:

```
symconfigure -sid SymmID -file set_device_dyn_rdf.cmd
-nop -v commit
```

To view the set of devices currently available for use in dynamic RDF operations:

```
symdev list -dynamic
```

Sample output:

Symmetrix ID: 000000005605

Device Name		Directors			Device		
Cap	Physical	SA	:P	DA :IT	Config	Attribute	Sts (MB)
000A	Not Visible	?????	02B:D0	Unprotected	Grp'd	RW	516
000B	Not Visible	?????	15A:D0	Unprotected	Grp'd	RW	516
000C	Not Visible	?????	01B:D0	Unprotected	Grp'd	RW	516
000D	Not Visible	?????	16A:D0	Unprotected	N/Grp'd	RW	516
000E	Not Visible	?????	02A:D0	Unprotected	N/Grp'd	RW	516
0012	Not Visible	?????	01B:C1	Unprotected	Grp'd	RW	516
0013	Not Visible	?????	16A:C1	Unprotected	Grp'd	RW	516
0014	Not Visible	?????	02A:C1	Unprotected	Grp'd	RW	516
0015	Not Visible	?????	15B:C1	Unprotected	N/Grp'd	RW	516
0016	Not Visible	?????	01A:D1	Unprotected	N/Grp'd	RW	516

For information on using the devices with RDF commands, refer to the *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.

Setting RDF Group Attributes

RDF group attributes allow you to assign priorities to SRDF/A sessions, and to set the minimum amount of time before attempting an SRDF/A cycle switch.

To set RDF group attributes, use the following command:

```
set [ra | rdf] group GroupNum
    [session_priority = SessionPriority] |
    [minimum_cycle_time = CycleTime];
```

where:

GroupNum = the RDF (RA) group number.

session_priority = the priority used to determine which SRDF/A sessions to drop if cache becomes full. Values range from 1 to 64, with 1 being the highest priority (last to be dropped).

minimum_cycle_time = the minimum time to wait before attempting an SRDF/A cycle switch. Values range from 5 to 59 seconds.

Example: Setting Session Priorities

To set the SRDF/A session priorities for two different RA groups, enter:

```
symconfigure commit -sid 12345 -file setup_rdfa.cmd
```

where `setup_rdfa.cmd` contains:

```
set ra group 24, session_priority=1;
set ra group 42, session_priority=8;
```

Configuring Meta Devices

You can form or dissolve a meta device, or add or remove a meta member. To initially create a meta member device, the head and members must be specified in the same command file. You can add or remove meta members from an existing concatenated meta device, or add members to a striped meta device. You can also dissolve a specified meta device, which subsequently releases all its members for normal use as stand-alone devices.

Creating a Meta Device

As mentioned in the introduction, when you initially create a meta device, you must form the head and add the members in the same command file. If using the `count` option, no members need to be specified. The configuration server will automatically select members from the pool of unmapped devices that match the meta head in size, emulation and configuration type.

First, to form a meta device, use the following command:

```
form meta from dev SymDevName,
    config=MetaOption
    [, stripe_size=<MetaStripeSize>[cyl]]
    [, count=<member_count>];
```

where:

config = the meta configuration type. Possible values are concatenated or striped.

stripe_size = the size of the striped meta device. This value can be expressed in blocks or cylinders. Possible sizes in 512 byte blocks are 1920, 3840, 7680, 15360, 30720, and 61440. If you do not specify a stripe size when creating a striped meta, two cylinders will be assigned. For information on meta volume strip sizes, refer to *Striped Data* on page 1-7.

cyl = if the `stripe_size` is expressed in cylinders, include `cyl` with the `MetaStripeSize` value. The size of a cylinder for FBA emulation is 960 512-byte blocks.

count = the total number of devices for the configuration server to add to the new meta device, including the head. This option may be omitted if you want to specify the members using the `add dev` command.

Enable SYMAPI Environment Option

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the `options` file), since meta devices must be unmapped for this operation. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Adding Meta Member(s)

Next, add the command specifying the new members.

Note: Omit this step if the `count` option was used previously with the `form meta` command.

```
add dev SymDevName[:SymDevName] to meta SymDevName;
```

Example: Forming a Concatenated Meta

To form a concatenated meta device 0010 and add meta members, enter:

```
form meta from dev 0010, config = concatenated;

add dev 0011:0012 to meta 0010;
```

Example: Forming a Striped Meta

To form a striped meta device 0010 and add meta members, enter:

```
form meta from dev 0010, config = striped,
    stripe_size=3840;

add dev 0013:0014 to meta 0010;
```

Note: It is recommended that you add all the meta members in the same session (same command file) where you form a striped meta device. You must add at least one meta member in the same session where you form a concatenated device.

To create a striped meta device by requesting the configuration server to form the meta device with four members; the meta head 0030 and three other members selected from the pool of unmapped devices, enter the following:

```
form meta from dev 0030, config = striped,
    stripe_size=1920, count=4;
```

Restrictions

The following are restrictions when configuring meta devices:

- ◆ The devices must be unmapped.
- ◆ Meta members cannot be removed from a striped meta device.
- ◆ To create a striped meta device, all members must be the same size and have the same mirror protection.
- ◆ To remove a member from a concatenated meta device, the member must be the tail. You cannot remove inner members.
- ◆ Only the head of a meta may have its type converted.
- ◆ Only the head of a meta is mapped or assigned to the host.
- ◆ All metas must contain at least one meta member. When a meta is formed, at least one member must be added.
- ◆ All meta devices must be composed of devices of the same BCV type (you must convert from RDF devices in order to change metas).
- ◆ Meta devices must be composed of devices that are all of the same FBA emulation type.
- ◆ You must create CKD meta devices using the `create dev` command, not the `form dev` command.
- ◆ VDEV meta devices must be composed of virtual devices that are of the same capacity, stripe size, count, and emulation type as the standard devices with which they will be used.
- ◆ An RDF device cannot be used to form a meta device (refer to Table 2-2 on page 2-10 for converting a meta device for RDF operation). The configuration manager cannot form a meta device from RDF devices that are RDFA-protected (that is, you cannot convert these devices to non-RDF, form the meta device, and convert them back to RDF).

Expanding an Existing Meta Device

Once you have successfully formed an initial meta with meta members, you can add additional meta members as necessary. The following sections show how to add an additional meta member or members to an existing concatenated or striped meta device.

Adding Members to a Concatenated Meta

To add additional members to an existing concatenated meta device, use the following form:

```
add dev SymDevName[:SymDevName] to meta SymDevName;
```

Example To add Symmetrix devices 0013 and 0014 to concatenated meta 0010, enter:

```
add dev 0013:0014 to meta 0010;
```

Adding Members to a Striped Meta

To add additional members to an existing striped meta device, use the following form:

```
add dev SymDevName[:SymDevName] to meta SymDevName
[, protect_data=[TRUE | FALSE],
  bcv_meta_head=SymDevName];
```

where:

protect_data = possible values are TRUE or FALSE. The **protect_data** option is only for striped metas. When set to true, the configuration manager automatically creates a protective copy to the BCV meta of the original device striping. Because this occurs automatically, there is no need to perform a BCV establish. When enabling protection via the **protect_data** option, you must specify a BCV meta identical to the existing (original) striped meta.

bcv_meta_head = when adding new members to an existing, striped meta device, if the data on the meta device is to be protected, you must specify the name of a **bcv_meta** that matches the original meta device in capacity, stripe count, and stripe size.

Example To add Symmetrix devices 0013 and 0014 to striped meta 0010, enter:

```
add dev 0013:0014 to meta 0010,
  protect_data=TRUE,
  bcv_meta_head=00CA;
```

Removing Meta Members

To remove a meta member or members from the tail of a concatenated meta device, use the following form:

```
remove dev SymDevName[:SymDevName] from meta
SymDevName;
```

Example To remove Symmetrix device 0014 from meta 0010, enter:

```
remove dev 0014 from meta 0010;
```

Dissolving a Meta Device

To dissolve a meta device, use the following form:

```
dissolve meta dev SymDevName;
```

Example To dissolve meta device 0010, enter:

```
dissolve meta dev 0010;
```

Note: A dissolve action removes the meta head and all its members.

Converting a Meta Device

To convert a meta device's configuration from concatenated to striped, use the following command:

```
convert meta SymDevName config=MetaOption>
[, stripe_size=MetaStripeSize>[cyl]],
[, protect_data=[TRUE|FALSE],
[bcv_meta_head=SymDevName];
```

where:

stripe_size = the stripe size in 512 byte blocks.

protect_data = specifies whether the data needs to be protected. Possible values are TRUE or FALSE.

bcv_meta_head = the Symmetrix device name.

Meta devices can be converted from concatenated meta devices to striped devices, and vice versa using the `convert meta` command. The command can also be used to change the stripe size of a striped meta device. Use the `protect_data` option to preserve data on a BCV meta device during conversion, and the `bcv_meta_head` option to specify which BCV meta device to use.

Example To convert a concatenated meta device 0030 to a striped meta device and preserve data during the conversion using BCV meta device 01F, enter:

```
convert meta 0030, config=striped, stripe_size=1920,  
protect_data=TRUE, bcv_meta_head=01f
```

Note: The specified BCV meta device must be identical to the original meta device in capacity, stripe count, and stripe size.

Data Preservation Concerns

When managing meta devices, data is not written directly to the devices involved, and therefore the original data is, essentially, preserved. However, due to the reorganization upon completion of the device management actions, the host is unable to access all of the original data. Therefore, it is the responsibility of the attached host to handle the change to the physical geometry of the device.

Note: The expansion of striped meta devices is an exception to this rule. In this case, you can provide a BCV meta to preserve the original data, which will be rewritten to the newly configured striped meta.

Table 2-3 lists the possible data management operations and their effects on data integrity.

Table 2-3 Meta Device Data Preservation

Meta Device Type with Data	Operation	Data Integrity Preserved Status
Meta devices concatenated	Adding a member	Preserved
Meta devices concatenated	Removing a member	Preserved ^a
Meta devices concatenated	Dissolving to non meta	Not preserved ^b
Meta devices concatenated	Converting to striped	Configurable ^c
Meta devices striped	Adding a member	Configurable ^c
Meta device striped	Dissolve	Not preserved ^b
Meta device striped	Converting to concatenated	Configurable ^c

a. Preserved up to where the meta device is removed.

b. Preserved but there is no host component to piece the data together.

c. Protection is a user-configurable option.



CAUTION

Dissolving striped or concatenated devices will lose access to data. Adding to concatenated meta devices preserves the meta device data, but makes the new members's original data inaccessible. Removing concatenated members will lose access to the data on the removed member. Use these commands with great caution.

Managing CKD Devices

You can map or unmap a range of CKD devices from an EA/EF port. In addition, you can also assign PAV alias addresses for devices already mapped.

Mapping a Range of Devices

To map a range of devices to an EA/EF port, use the following form:

```
map dev SymDevName[:SymDevName] to dir
    DirectorNum:PortNum, starting
    base_address=cuu_address [mvs_ssid=nnn];
```

To map a range of devices to an EA or EF port, using the same addressing from a different port, use the following form:

```
map dev SymDevName[:SymDevName] to dir
    DirectorNum:PortNum, copying dir
    DirectorNum:PortNum;
```

where:

base_address = The base address to be assigned to the first device in the mapping request. It will be incremented by one for each device in the range of devices being mapped.

Mainframe ports expect devices to be mapped in groups to form CU images. The first digit in the address is the CU image number, which can range from 0 to 0xF. The remaining two digits can range from 00 to 0xFF.

You can divide the set of 256 cu_addresses available in a CU image into base addresses and aliases, as is required.

mvs_ssid = When mapping a range of devices to an EA/EF port, you may need to change the current *mvs_ssid* assigned to the devices. If the devices are becoming part of an existing CU image, they will be assigned the *mvs_ssid* of the devices already mapped. If a new CU image is being formed and mapped, a new *mvs_ssid* can be assigned during the map request. It is not valid to have some devices with a particular SSID mapped and some unmapped.

Example To map devices 01D through 023 to CU image A, where the next free base address is 20, enter:

```
map dev 01D:023 to dir 01C:1 starting base_address=A20;
```

Unmapping a Range of Devices

Unmapping can be from one or all ports. Since all devices with the same SSID must either be mapped or unmapped, you must provide an SSID when unmapping only some devices in a CU image.

To unmap a range of devices from EA or EF ports, use the following form:

```
unmap dev SymDevName:[SymDevName] from dir
< ALL:ALL | ALL:PortNum | DirectorNum:ALL |
DirectorNum:PortNum [new_ssid=nnn];
```

Example To unmap five devices from a CU image so that meta CKDs can replace them, and assign the devices an SSID different from the CU image, enter:

```
unmap dev 13B:13F from dir all:all, new_ssid=0160;
```

Copying the Mapping of a Range of Devices

To copy a CU image mapping, all devices in the range must be mapped to the same CU image, or not mapped at all. Devices within the specified range that are not mapped will be ignored as long as they are not mappable (SAVE devices, DRVs, etc.). If a device in the range is mappable, the request will be rejected.

To copy a CU image mapping from one port to another, use the following form:

```
map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNum copying dir DirectorNum:PortNum
```

Example To copy addresses of devices 010 through 050 from port 03C:0 to port 03D:0, enter:

```
map dev 010:050 to dir 03D:0 copying dir 03C:0;
```

Assigning PAV Alias Addresses to Mapped Devices

When assigning PAV alias addresses to mapped devices, the aliases will be propagated to all director ports to which the devices are mapped. Devices within the range that are not mapped will be skipped. If any devices in the range are mapped to a different CU image than the first device, an error will be returned. If the device range has base addresses with gaps, the aliases will also have gaps.

To assign PAV alias addresses to devices mapped to EA or EF ports, use the following form:

```
add pav alias to dev SymDevName[:SymDevName] starting  
alias=cuu_address;
```

where:

alias = An alternative front-end mapping address, used by a mainframe host to access a device.

Mainframe ports expect devices to be mapped in groups to form CU images. The first digit in the address is the CU image number, which can range from 0 to 0xF. The remaining two digits can range from 00 to 0xFF.

Example To add the alias A60 to device 01D, which is already mapped, enter:

```
add pav alias to dev 01D starting alias=A60
```

Removing PAV Alias Addresses From Mapped Devices

To remove PAV alias addresses from devices mapped to EA or EF ports, use the following form:

```
remove pav alias from dev SymDevName[:SymDevName],  
starting alias=cuu_address;
```

Example To remove aliases from CU image 7 so that additional devices can be mapped to it, enter:

```
remove pav alias from dev 012:052, starting alias=750;  
remove pav alias from dev 012:052, starting alias=790;
```

Managing SAVE Device Pools

Beginning with Enginuity Version 5671 TimeFinder/Snap uses SAVE device pools to store pre-update images or change tracks during a virtual copy session. Symmetrix supports the creation of multiple named SAVE device pools, allowing `symsnap` commands to select a particular pool. This alleviates contention for SAVE device space among several Snap sessions and eliminates the possibility of a single session using all pool space.

This section explains how to create and delete SAVE device pools, add and remove devices from pools, as well as enable and disable their members.

To list and show pools and their content, use the `symsnap` command.

Creating a SAVE Device Pool

To create a SAVE device pool, use the following form:

```
create pool PoolName, type=savedev;
```

where:

PoolName = the name of the SAVE device pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-), and underscore (_) characters. The name `DEFAULT_POOL` is reserved to represent the container of all *unpooled* devices.

New SAVE device pools are disabled at creation. To enable the pool, you must add and enable one or more SAVE devices in the pool.

The maximum number of SAVE device pools per Symmetrix is 256.

Example

To create a new pool named *Finance*, enter:

```
create pool Finance, type=savedev;
```

Deleting a SAVE Device Pool

You can only delete a SAVE device pool whose members are disabled and inactive. Once you delete the SAVE device pool, its members will become part of the default pool.

To delete a SAVE device pool, use the following form:

```
delete pool PoolName, type=savedev;
```

Example

To delete a pool named *Finance*, enter:

```
delete pool Finance, type=savedev;
```

Adding SAVE Devices to a Pool

You can only add disabled and inactive SAVE devices to a pool. While adding the devices, you can specify whether they should be enabled or disabled in the pool.

To add SAVE devices to a pool, use the following form:

```
add dev SymDevName[:SymDevName] to pool PoolName
    type=savedev [, member_state=ENABLE | DISABLE];
```

where:

member_state = specifies whether the SAVE device(s) being added should be enabled or disabled in the pool. Possible values are ENABLE or DISABLE.

Example

To add SAVE devices to a pool named *Finance*, enter:

```
add dev 001B:001D to pool Finance, type=savedev,
    member_state=ENALBE;
```

Removing SAVE Devices From a Pool

You can only remove disabled and inactive SAVE devices from a pool. Once you remove a device from a pool, it will become part of the default pool.

To remove SAVE devices from a pool, use the following form:

```
remove dev SymDevName[:SymDevName] from pool PoolName,
    type=savedev;
```

Example

To remove a SAVE device from a pool named *Finance*, enter:

```
remove dev 001D from pool Finance, type=savedev;
```

Enabling/Disabling SAVE Devices

You can enable or disable SAVE devices for use in a pool. The devices in a pool do not all have to be in the same state (enabled or disabled). If all the devices in a pool are disabled, then the pool is disabled. If at least one device in a pool is enabled, then the pool is enabled. To enable or disable a range of devices, all the devices must be in the same pool. To disable a device, all sessions must be terminated, and have no used tracks.

To enable or disable a SAVE device, use the following form:

```
enable | disable dev SymDevName[:SymDevName] in pool  
    PoolName, type=savedev
```

where:

PoolName = the name of the SAVE device pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-), and underscore (_) characters.

Example To enable the use of three new SAVE devices in a pool named *Finance*, enter:

```
enable dev 01B:01D in pool Finance, type=savedev;
```

Setting Port Characteristics

You can set the port characteristics of a specified director using the following command:

```
set port DirectorNum:PortNum
  [FlagName=enable|disable
  [, FlagName=enable|disable]]
gige primary_ip_address=IPAddress
      primary_netmask=IPAddress,
      default_gateway=IPAddress
  [fa_loop_id=integer] [hostname=HostName];
```

where:

default_gateway = the gateway or router address for a front-end Gig-E port.

FlagName = a SCSI or fibre port flag. Possible values for the SCSI protocol flags are in Table 2-4 on page 2-46, and the values for the fibre protocol flags are in Table 2-5 on page 2-48.

fa_loop_id = the FA director loop ID (arbitrated loop physical address). (0 - 125) (Hard Addressing must be enabled.)

gige = indicates that one or more network address values are going to be specified for a front-end Gig-E director. Addresses should use the Internet standard dot notation.

hostname = 12-character hostname.

primary_ip_address = the IP address for a front-end Gig-E port.



CAUTION

When changing the primary IP address, be sure to update all connected hosts with the new address.

primary_netmask = the IP netmask for a front-end Gig-E port.



CAUTION

Incorrectly changing the port flags can render your storage system inaccessible. Be sure of your needs before resetting these flags.

Table 2-4 SCSI Protocol Port Flags

SCSI Protocol Flags	Description
Auto_Busy ^a	When enabled specifically for Unisys A-series platforms only, this flag enables the auto-busy mechanism so that the Symmetrix array returns a Busy to all Unisys host requests.
Avoid_Force_Negotiate ^a	When this flag is enabled for Sequent V4.2.3 and lower, the Symmetrix array never initiates negotiations. Normal Symmetrix behavior is to initiate negotiations after an offline-to-online transition. This is for hosts that do not handle negotiations.
Avoid_Reset_Broadcast	When this flag is enabled, a SCSI bus reset only occurs to the port that received the reset (not broadcast to all channels).
Command_Reordering ^a	When this flag is enabled with Tag Command Queueing in use, the incoming SCSI commands become reordered to Simple Queueing. The default is enabled and should only be disabled upon a request from EMC.
Common_Serial_Number	This flag should be enabled for multipath configurations or hosts that need a unique serial number to determine which paths lead to the same device.
Cyl_Count_In_Name ^{a,b}	When this flag is enabled, the Symmetrix via the specified port embeds the cylinder count into the product ID returned in the SCSI Inquiry command. Enabled for Pyramid only when it is desirable to embed the Symmetrix support into the Pyramid kernel.
Disable_False_Disconnect ^{a,b}	When enabled for debugging, this flag prevents the port from performing a False Disconnect operation. (Default is disabled and currently, you cannot change this flag.)
Disable_Interleaved_Cmds ^a	When enabled (always), meta volume command interleaving is being supported. This allows multiple meta members to operate at the same time on the same volume.
Disable_Mini_Q ^{a, b}	When enabled for debugging, this flag disables the use of the Mini Queue on the port. (Default is disabled and currently, you cannot change this flag.)
Disable_Q_Reset_on_UA	When this flag is enabled, a Unit Attention (UA) that is propagated from another director does not flush the queue for this device on this director. Used for hosts that do not expect the queue to be flushed on a 0629 sense (only on a Hard Reset).
Disable_Ultra ^{a,b}	When enabled, this flag disables Ultra SCSI on an Ultra capable SA port. (Default is disabled and currently, you cannot change this flag.)
Environ_Set	When enabled, this flag enables the environmental error reporting by the Symmetrix to the host on the specific port.
HP3000_Mode	When enabled for HP MPE 5.0 and Enginuity Version 5062 and lower, this flag causes the Symmetrix port to return a SCSI Busy state instead of a 0B44 sense code when a xx3C error occurs.
Linked_Commands ^{a,b}	When enabled, this flag enables support of SCSI linked commands. It allows a host to chain SCSI commands in a manner similar to mainframe Channel Command Words (CCWs). (Default is enabled, and currently, you cannot change this flag.)

Table 2-4 SCSI Protocol Port Flags (continued)

SCSI Protocol Flags	Description
Negotiate_Reset ^c	When enabled for AS/400 hosts, this flag forces a SCSI negotiation by the Symmetrix array after a: • SCSI reset • Error • Bus device reset This flag is used for AS/400 systems only (default is off).
PBAY_Monitor ^a	For the Sequent platforms only to allow emulation of the Sequent PBAY. When enabled, this flag enables low-level polling of the SCSI bus in order to intercept the nonstandard SCSI operations required for a Sequent PBAY disk subsystem. Must be used for the Sequent cluster operation for the Symmetry system for Sequent V4.2.x operating systems only. Must not be used on versions higher than V4.2.x or for any NUMA-Q systems and also not used for Fibre Channel.
SCSI_3	When enabled, the Inquiry data is altered when returned by any device on the port to report that the Symmetrix supports SCSI 3 protocol. When this flag is disabled, the SCSI 2 protocol is supported.
Sequent	When enabled, the Symmetrix on the specified port does not force wide or synchronous negotiations, and sets the task timeout window to be 15 seconds before aborting a process. In addition, a busy status is returned instead of an 0B44H when aborting a command on a timeout.
Server_On_AS400	When enabled for AS/400 platforms, this flag indicates that the port is to behave as a server, returning server inquiry data, rather than AS/400 data.
Set_Qerr ^a	This flag should be enabled for SGI platforms only to flush the queue on a contingent allegiance condition (CAC). Must be used for V5.3 and V6.2 SGI operating systems and cluster environments. Not used on versions later than V6.2.
Siemens	For Siemens R-Series platforms only. When this flag is enabled for Siemens, the Symmetrix array returns in the sense data error 0B48, instead of 0B44 for normal behavior.
Soft_Reset	When enabled for a Bull/GCOS-7 host, the Symmetrix port supports the SCSI Soft Reset option.
Sunapee	When enabled for Sun PDB clusters, this flag enables the Sunapee option on the port.
Sync_Transfer ^{a,b}	When enabled, this flag enables SCSI synchronous negotiations. (Default is enabled, and currently, you cannot change this flag.)
Tagged_Commands ^{a,b}	When enabled, this flag enables support for tagged commands. (Default is enabled, and currently, you cannot change this flag.)
Wide_Transfer ^{a,b}	When enabled, this flag enables SCSI Wide operation. (Default is enabled, and currently, you cannot change this flag.)

a. Obsolete in Enginuity Version 68 and higher.

b. Not available for host-based configuration changes.

c. Can only be set on SA ports with Enginuity Version 5568 and higher.

Table 2-5 Fibre Protocol Port Flags

Fibre Protocol Flags	Description
Class_2_Service ^{a,b}	Enabled for a Class 2 fibre protocol connection that requires an acknowledgement for each frame transmitted. (You cannot change this flag .)
Disk_Array ^a	When enabled (default), the port is represented as a disk array. This port information appears in the Inquiry data.
Generic_VSA ^{a,b}	When enabled, the generic volume set addressing mode is selected. GVSA mode allows hexadecimal addressing. (You cannot change this flag .)
Global__3rdParty_Logout ^a	When enabled (default) for Engenuity Version 5x65 and higher, an extension is provided to the existing third-party logout required by the standard. In addition to logging out the hosts who are logged in to the port receiving the third-party logout, the logout propagates to other Symmetrix fibre ports that share volumes with the port that had received the logout.
Hard_Address ^a	When enabled (default), the FA director attempts to get the loop_id specified when it initializes on the loop (hard-assigned addressing). When disabled, soft addressing is being used.
Init_Point_to_Point ^c	When enabled for Engenuity Version 5x65 and higher at configuration time, specifies a point-to-point (direct or switched) topology in the initialization sequence. When disabled (default), it is initialized as an arbitrated loop.
Non_Participating ^c	When enabled along with the Hard_Address flag, the Fibre Channel director only uses hard-assigned addressing when it initializes on the loop. Otherwise, soft-assigned addressing is used during loop initialization (the default).
OpenVMS	Enabled for an OpenVMS fibre connection.
Unique_WWN ^c	When enabled (default) for all environment configuration changes and new environments to ensure unique World Wide Names (WWN) within the fibre environment (uses Symmetrix serial numbers and port numbers). When disabled, you don't have to change WWNs.
VCM_State	Enabled for device masking or the Volume Logix software, which provides volume configuration management controls to handle access to Symmetrix devices. (Disabled is the default.)
Volume_Set_Address ^a	When enabled along with the Disk_Array flag for HP-UX hosts, the volume set addressing mode is selected. VSA mode allows octal addressing.

a. Obsolete in Engenuity Version 68 and higher.

b. Not available for host-based configuration changes.

c. Not available for Gig-E ports.

Example: Setting Port Characteristics

To turn on hard-assigned addressing and set the FA loop ID to 3 for director 03A, port 0, enter:

```
set port 03A:0 Hard_Address=enable, fa_loop_id=3;
```

Impact on I/O

When setting port attributes, it is recommended that you temporarily suspend I/O activity to the effected ports during this operation.

Enable SYMAPI Environment Option

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if there is no device from the local host mapped to this port. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Viewing Port Flags

You can display the possible port flags and their current status with the following command:

```
symcfg -SA all [-P #] -v list
```

The following is a partial example of output:

```
Director Identification: SA-14B
  Director Type           : SCSI
  Director SCSI Capability : N/A
  Director Status         : Online

  Number of Director Ports : 2
  Director Port Status      : [ON,ON,NA,NA]

  Director Symbolic Number : 14B
  Director Numeric Number   : 30
  Director Slot Number      : 14

Port: 0

  SCSI Flags
  {
    Tagged_Commands      : Enabled
    Linked_Commands      : Enabled
    Sync_Transfer        : Enabled
    Wide_Transfer        : Enabled
    Negotiate_Reset      : Disabled
    Soft_Reset           : Disabled
    .
    .
    .
```

Setting Symmetrix Metrics

You can set some attributes to gain access to new Symmetrix features using the following command:

```
set Symmetrix MetricName=MetricValue
[, MetricName=MetricValue];
```

where:

MetricName = the Symmetrix metric to be set. Possible name values are:

`concurrent_rdf` — Enables access to the concurrent RDF feature - multiple R2 devices for a single static R1. Possible values are `ENABLE` or `DISABLE`.

`dynamic_rdf` — Enables the creation of a set of devices that are RDF-capable, (can be dynamically assigned as RDF1 or RDF2 devices). Possible values are `ENABLE` or `DISABLE`.

`dynamic_concurrent_rdf` — Enables access to the concurrent RDF feature (multiple R2 devices for a single R1) for dynamic RDF devices. Possible values are `ENABLE` or `DISABLE`.

`fba_multi_access_cache` — Determines whether a read request can share cache slots in some conditions. Possible values are `ENABLE` or `DISABLE`.

`max_hypers_per_disk` — Specifies the maximum number of hypers that can be created on a physical disk. Possible values are 1 to 255, where the upper limit varies with the Engenuity level.

`max_pav_aliases` — If PAV is enabled, specifies the maximum number of aliases that can be assigned to a devices. Possible values for a Symm 6 are 1-7 using 5670, 1-15 using 5671.

`PAV_mode` — Enables the use of PAV (Parallel Access Volumes). Possible values are:

`STANDARD` — Standard PAV volumes with static aliasing can be configured.

`DYNAMIC_STANDARD` — Standard PAV volumes with dynamic aliasing can be configured.

`rdfa_cache_percent` — The percentage of write pending cache that can be used by RDFA. This is a value from 0 to 100 percent.

`rdfa_host_throttle_time` — The number of seconds to throttle host writes to SRDF/A devices when cache is full, before dropping RDFA sessions. Throttling will delay a write from the host until a cache slot becomes free. Values are from 0 to 65535.

`raid_5_support` — Enables the ability to create RAID-5 devices in the Symmetrix. Possible values are `ENABLE` or `DISABLE`.

`raid_s_members` — If RAID-S support is enabled, this identifies the number of members in the Parity RAID set. Possible values are 3 or 7. This also controls the RAID-5 member count.

`raid_s_support` — Enables the ability to create RAID-S or parity RAID devices in the Symmetrix array. Possible values are `ENABLE` or `DISABLE`.

`VCMDB_restricted_access` — When enabled, restricts host access to the VCM database as indicated by the user. Possible values are `ENABLE` or `DISABLE`.

Example To change the maximum number of hypervisors that can be created in the Symmetrix array from 16 to 32, enter in the command file:

```
set Symmetrix max_hypers_per_disk=32;
```

Viewing the Symmetrix Metrics

You can view the current settings for these metrics with the following command:

```
symcfg -sid SymmID -v list
```

Swapping RA Groups

You can swap all devices in an RA group from target to source, using the following command:

```
swap ra group nnn,
  refresh=R1|R2, start_copy=YES|NO;
```

where:

refresh = the RDF device (R1 source or R2 target) to refresh.

start_copy = whether an RDF pair should be synchronized after the configuration change is committed.

Example: Swapping RA Group

To swap RA group 1 from the R1 source to an R2 target group, and then refresh the R2 device, enter:

```
swap ra group 1, refresh=r2, start_copy=yes;
```

Refresh Data Concerns

The refresh action identifies which device does **not** hold a valid copy of the data before the swap operation begins. If you determine that the R1 holds the valid copy, the action of `refresh R2` will obtain a count of the tracks that are different on the R2 and will copy (refresh) these tracks from the R1 to the R2 device. The result will be the reverse if you use `refresh=R1` as the action.

Restrictions and I/O Impact

The following are restrictions on swapping RA groups:

- ◆ All the devices in the RA group must be of the same type (RDF1 or RDF2) if FarPoint™ is enabled.
- ◆ Only one RA group may be swapped per configuration session.
- ◆ When swapping source and target attributes—no restriction on I/O to R2, but no I/O allowed to the R1 device.
- ◆ When swapping the RA group personalities that engage ESCON directors in a FarPoint connection, be aware that FarPoint buffer settings cannot be adjusted using `symconfigure`. If your FarPoint buffers are set to customized parameters other than default values, an EMC representative will need to be called to adjust the buffer settings after the swap has taken place.

For information about swapping R1/R2 device designations, refer to *Swapping R1 Devices with R2 Devices* on page 2-56.

**Enable SYMAPI
Environment Option**

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if the devices are not mapped to the user host. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Managing Spare Disks

You can reserve physical disks as dynamic (hot) spares. When a physical disk is reserved as a dynamic spare it is not allocated to any device and is held in reserve to support the hypers of a Symmetrix disk that has failed.

Adding a Spare Disk

A spare disk can only be created from a new disk containing no hypers. To reserve a disk as a dynamic spare, use the following command:

```
create spare count=n[, format = [512 | 520]];
```

where:

count = a positive integer the defines the number of disks to be set aside as spares.

format = the recording format to be used on a spare disk. Values are 512 or 520. Select a format based on the type of disk it should be able to replace.

- For Symmetrix DMX models:
 - for CKD and FBA use 512
 - for AS/400 and Tandem use 520
- For Symmetrix 5 models:
 - for CKD and FBA use 512
 - for AS/400 and Tandem use 520
- For Symmetrix 4 models:
 - With selective LLF enabled:
 - for CKD and FBA use 512
 - for AS/400 and Tandem use 520
 - If no AS/400 or Tandem devices:
 - for CKD and FBA use 512
 - If AS/400 or Tandem devices:
 - for CKD use 512
 - for FBA, AS/400, and Tandem use 520

Deleting a Spare Disk

To remove a disk as a dynamic spare, making it available for normal storage, use the following command:

```
delete spare disk=[director_num, da_interface,  
                   scsi_id];
```

where:

director_num = The director identity number, such as 16A.

da_interface = The DA SCSI path (c, d, e, or f).

scsi_id = The SCSI target ID (decimal value).

For example:

```
delete spare disk=[16A, D, 0];
```

Viewing Existing Spares

To view the set of existing spares, use the following command:

```
symdisk list -hotspare
```

Impact on I/O

The following are restrictions/conditions in lieu of impact on I/O activity:

- ◆ When adding or removing Symmetrix spares—no restriction on I/O.
- ◆ A spare cannot be removed while in use.

Swapping R1 Devices with R2 Devices

You can swap the RDF personality of the RDF device designations of a specified device group, if you have any of the following:

- ◆ An SRDF license and a Configuration Manager license with Enginuity Version 5x66 or 5267.
- ◆ Or, an SRDF license and a Configuration Manager license with Enginuity Version 5567 or higher, and Dynamic RDF is disabled in your Symmetrix configuration.
- ◆ Or, an SRDF license with Enginuity Version 5567 or higher and Dynamic RDF is enabled in your Symmetrix configuration.

Source R1 device(s) become target R2 device(s) and target R2 device(s) become source R1 device(s).

Use the following form to perform an R1/R2 swap:

```
symrdf -g DgName [-force][-symforce] [-bcv|-all]
              [-bypass] [-v|-noecho] [-noprompt]
              [-i Interval] [-c Count]
swap [-refresh R1|R2]
```

For more information about this command, refer to the *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.

Note: R1/R2 swap operations are not supported for Enginuity Version 5669 or later configurations where the R2 device is larger than the R1 device.

Impact on I/O

The following are restrictions/conditions in lieu of impact on I/O activity:

- ◆ When swapping source and target attributes, there are no restrictions on I/O to the R2 device, but no I/O is allowed to the R1 device.
- ◆ When swapping the RA group personalities that engage ESCON directors in a FarPoint connection, FarPoint buffer settings cannot be adjusted using `symconfigure`. If your FarPoint buffers are set to customized parameters, other than default values, an EMC Representative will need to be called to adjust the buffer settings after the swap has taken place.

Swap Preparation

If your RDF devices are enabled for dynamic swap with Enginuity Version 5x67, ensure that both the R1 and R2 devices are in the same state, either failed over or suspended with both sides write-disabled to the source host, before attempting a swap.

Enable SYMAPI Environment Option

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if the devices are not mapped to the user host. For information about enabling this option, refer to *Enable SYMAPI Environment Option* on page 2-59.

Stages of the Configuration Change Action

A lock on the Symmetrix configuration is engaged by the Symmetrix configuration server during the configuration change session. Note that it does not lock out normal I/O activity, it simply blocks others from attempting to change the configuration.

A session acting on a specified `symconfigure` command file is processed in a series of progressive stages that can be monitored using polling options:

- ◆ preview
- ◆ prepare
- ◆ commit

Not all stages are always executed. Use discretion when controlling which stages are to be completed to allow checking and debugging of the command file before the changes are implemented.

Preview Stage

The `preview` argument (stage) verifies the syntax and correctness of each individual change defined, their correctness as a set, and then terminates the session without change execution (correct within the realm of the host and valid as a possible Symmetrix configuration).

To preview the command file that creates a concatenated meta device in Symmetrix 12345, using device 0010 as the meta head, and 0011 - 0013 as members, enter:

```
symconfigure -sid 12345 preview -file add_meta.cmd
-noprompt -v
```

where `add_meta.cmd` contains:

```
form meta from dev 0010, config = concatenated;

add dev 0011:0013 to meta 0010;
```

Prepare Stage

The `prepare` argument (stage) will perform the preview checks and also verify the appropriateness of the resulting configuration definition against the current state of the Symmetrix arrays; the argument then terminates the session without change execution (correct within the realm of both the host and the Symmetrix array). The `prepare` argument has no function for SAVEDEV pool sessions.

To prepare the command file that adds four new Symmetrix devices to Symmetrix array 000000012345 as two-way mirrored devices with a size of 1100 cylinders, using FBA emulation, enter:

```
symconfigure -sid 12345 prepare -file
  add_new_symdevs.cmd -noprompt -v
```

where `add_new_symdevs.cmd` contains:

```
create dev count=4, size=1100,
  emulation=FBA, config=2-Way-Mir;
```

Commit Stage

The `commit` argument (last stage) completes all stages and activates the changes in the specified Symmetrix array.

Impact on I/O

Some classes of change operations might impact current I/O. Consider the *Impact on I/O* statement for each class operation in this chapter before working on any online devices. When possible, before you `commit` any action, stop I/O activity on the Symmetrix devices to be altered for a configuration change session.

For example, set the device(s) for change to be Not Ready using the following command type:

```
symdev -sid SymmID not_ready SymDevName
```

Enable SYMAPI Environment Option

Before executing any of the class changes, environment option `SYMAPI_CTRL_OF_NONVISIBLE_DEVS` in the `options` file might need to be enabled or removed. This option or constant is normally enabled, which is required for many of the configuration changes available. (For specific impact, see the reference to this option in each of the Class descriptions.)

Note: A nonvisible device is a device that is not mapped to your host and/or does not have a physical host device name.

The editable `options` file is located in the SYMAPI configuration directory:

```
/var/symapi/config (for UNIX)
```

```
C:\Program Files\EMC\Symapi\config (for Windows)
```

To enable `SYMAPI_CTRL_OF_NONVISIBLE_DEVS`, set:

```
SYMAPI_CTRL_OF_NONVISIBLE_DEVS=ENABLE
```

Commit Example

To commit the command file that creates a concatenated meta device in Symmetrix 12345, using device 0010 as the meta head, and 0011 - 0013 as members, enter:

```
symconfigure -sid 12345 commit -file add_meta.cmd  
-noprompt -v
```

where `add_meta.cmd` contains:

```
form meta from dev 0010, config = concatenated;  
add dev 0011:0013 to meta 0010;
```

Monitoring the Status of the Change Session

After you commit a command file for execution, it will take a few minutes to complete execution. You can check the status of any configuration change session using the `query` argument.

To check the status of the change session for the next 2 minutes, every 10 seconds on Symmetrix 12345, enter:

```
symconfigure -sid 12345 query -i 10 -c 12
```

This command is useful in RDF environments, where a change to a local Symmetrix array on one host results in a corresponding change to a remote Symmetrix array. The System Manager of a host connected to the remote Symmetrix array can monitor the progress of the change. A `query` is also helpful at sites where the Symmetrix Optimizer is also modifying a configuration by rearranging the placement of a Symmetrix device's mirrors.

Aborting the Change Session

You can abort a change session. The `abort` action also frees the configuration session lock.

To abort a change session on Symmetrix 12345, enter:

```
symconfigure -sid 12345 abort
```

Note: Because changes made in the RDF operations class will initiate actions on the local Symmetrix and remote Symmetrix arrays, it might become necessary to abort processing on a remote Symmetrix array.

Note: At some point during `commit` processing, a *point of no return* is reached. Any attempt to abort will be denied once processing has reached this point.

Subsystem IDs for Mainframe Systems

Subsystem identifiers (SSID) are used by z/OS (MVS) mainframe systems to locate physical disk controllers. In a Symmetrix array that is configured to support only open systems hosts, the SSID value is not required. On Symmetrix arrays that service both mainframe (CKD devices) and open system hosts (FBA devices), the SSID parameter is required when creating new devices. The SSID parameter is also required when removing a mirror from a device, which results in the mirror becoming a new independent device.

Depending on the type of CKD emulation that is configured, the maximum number of SSID counts may change. It is recommended that you contact EMC Customer Support and ask which SSID was last used and what would be the next recommended ID to use along with how many devices can be placed behind it before you are required to increment the value to the next major range.

To obtain a list of SSIDs that are currently in use, use the following command:

```
symcfg list -ssid -sid SymmID
```


Updating the Host

After you reconfigure a Symmetrix array by moving, deleting, adding, or modifying one or more devices, you must update the host so that the host recognizes the new Symmetrix configuration.

For some platforms, the `symcfg scan` command is available to perform the host update. These include Sun Solaris, HP-UX, IBM AIX, Tru64/OSF1, and Windows systems.

This appendix describes the procedures for these host systems.

- Sun Solaris SystemsA-2
- HP-UX SystemsA-3
- IBM AIX SystemsA-4
- HP Tru64 UNIX Systems.....A-5
- Windows SystemsA-6

Sun Solaris Systems

Whenever devices are newly mapped to a host system, or device channel addresses are changed, you need to perform the following actions to introduce the new devices to the system.

To add new Symmetrix devices while online in the Solaris environment, perform the following steps:

1. Execute the following utilities:

```
drvconfig  
disks  
devlinks
```

2. Follow the instructions in the *Solaris* chapter of the *Open Systems Host Environment Product Guide* to introduce new devices to the host environment.

HP-UX Systems

Whenever devices are newly mapped to a host system or device channel addresses are changed, you need to perform the following actions to introduce the new devices to the system.

To view the results of the mapping changes, use the `ioscan` command in a statement similar to the following:

```
ioscan -fnC disk
```

You can define newly connected physical volumes to the HP-UX host system without rebooting it, using the following form:

```
insf -e
```

For more information, refer to the *HP 9000* chapters of the *Open Systems Host Environment Product Guide*.

IBM AIX Systems

Whenever devices are newly mapped to a host system or device channel addresses are changed, you need to perform the following actions to introduce the new devices to the AIX system.

1. From the **SMIT** menu, select **Devices, Fixed Disk, Add a Disk**.
2. Select the **EMC SYMMETRIX** definition from the disk table.
3. Select the SCSI bus on which the new disk resides.
4. Type the connection address for the new device (target, LUN).
5. Select **EXECUTE**.
6. Repeat steps 2 through 5 for each new device being added to the configuration.

HP Tru64 UNIX Systems

Whenever devices are newly mapped to a host system or device channel addresses are changed, you need to perform the following actions to introduce the new devices to the system.

When using Tru64 UNIX, you can introduce the new devices to the system with these steps:

1. At the prompt, type:

```
scsimgr -scan_bus bus=BUSNUM
```

2. Repeat for each LUN:

- a. Write a label to the device you are defining:

```
disklabel -rw rz<lun_letter><unitID> <label>
```

- b. Change the ownership on the device to a particular application:

```
chown <owner>:<group> *rz<lun letter><unitID>*
```

3. Follow the instructions in the *Tru64 UNIX* chapter of the *Open Systems Host Environment Product Guide* to introduce new devices to the host environment.

Windows Systems

Whenever devices are newly mapped to a host system or device channel addresses are changed, you need to perform the following actions to introduce the new devices to the system.

Windows 2000

To add new Symmetrix devices to a Windows 2000 system while the system remains online, perform these steps:

1. From the desktop, select **Start, Settings, Control Panel, Add /Remove Hardware**. Complete the wizard to discover and add the new Symmetrix devices.
2. Partition and format the new devices as described in the *Windows 2000* chapter of the *Open Systems Host Environment Product Guide*.

Windows 2003

To add new Symmetrix devices to a Windows 2003 system while the system remains online, perform these steps:

1. From the desktop, select **Start, Settings, Control Panel, Add /Remove Hardware**. Complete the wizard to discover and add the new Symmetrix devices.
2. Partition and format the new devices as described in the *Windows 2003* chapter of the *Open Systems Host Environment Product Guide*.

A

- abort argument 2-62
- attributes, setting 2-50

B

- BCV configurations 1-15
- BCV devices 1-15
- business continuance
 - operations 1-6

C

- change sessions
 - aborting 2-62
 - monitoring changes 2-61
- CKD devices 2-63
 - assigning PAV aliases 2-41
 - copying a CU image 2-40
 - mapping a range 2-39
 - removing PAV aliases 2-41
 - unmapping a range 2-40
- classes 2-3
- command file 2-7
- comments xiv
- commit argument 2-59
- configuration component 1-2
- configuration verification 2-5

D

- data, refreshing 2-52
- device locking 2-3
- device type
 - conversion 2-8

- creating 2-8
- devices
 - converting 2-19
 - creating (adding) 2-13
 - deleting 2-17
 - mapping 2-24
 - mapping to front-end ports 2-24
 - setting characteristics 2-22
 - unmapping 2-26
- disk management
 - adding a spare disk 2-54
 - deleting a spare disk 2-55
- disk space 2-5
- DRV devices 1-4
- Dynamic RDF 1-4, 2-28
- Dynamic RDF devices, configuring 2-28
- Dynamic swap 2-57

E

- environment option, non visible devices 2-59

F

- fibre protocols 2-45
- free disk space 2-5

H

- held devices 2-3
- host updating 2-6

M

- meta devices 1-6

- configuration restrictions 2-34
- configuring 2-36
- converting 2-36
- creating 2-10, 2-32
- dissolving 2-36
- meta head 1-6
- performance 1-7
- size 1-6
- meta members
 - adding concatenated 2-33
 - adding striped 2-33
 - removing 2-36
- mirror set protections 1-10
- mirrors
 - RAID-S 1-14
 - three way 1-13
 - two way 1-12
- MVS subsystem IDs 2-14, 2-63

N

- new devices, introducing to
 - AIX systems A-4
 - Digital UNIX systems A-5
 - HP-UX systems A-3
 - Solaris systems A-2
 - Windows systems A-6
- non visible devices 2-59

O

- operations classes 2-3

P

- parity RAID 1-13
- port characteristics, setting 2-45
- port flags 2-45
- prepare argument 2-58
- preview argument 2-58

Q

- query argument 2-61

R

- R1/R2 devices, swapping RDF personalities 2-56
- RA groups, swapping 2-52

- RAID-S 1-14
- RDF devices, converting from static to dynamic 2-19
- RDF group attributes, setting 2-31
- refresh option 2-52
- restrictions
 - for configuring meta devices 2-11, 2-34
 - for configuring SAVE devices 2-11
 - for configuring virtual devices 2-11
 - for converting devices 2-21
 - for creating devices 2-16
 - for deleting devices 2-17
 - for setting device characteristics 2-23
 - for swapping RA groups 2-52

S

- safe disciplines 2-5
- safety considerations 2-5
- sales locations xiv
- SAVE device pools
 - adding devices 2-43
 - creating 2-42
 - deleting 2-42
 - enabling/disabling devices 2-44
 - removing devices 2-43
- SAVE devices 1-9
- SCSI protocols 2-45
- SDR
 - mapping 2-24
 - unmapping 2-26
- sevice attributes, setting 2-22
- spare disks
 - adding 2-54
 - removing 2-55
 - viewing 2-55
- SRDF configurations 1-15
- SRDF devices 1-15
- SRDF/A sessions, assigning priority 2-31
- SSIDs 2-63
- stripe sizes 1-8
- striped data 1-7
- striped meta
 - protect_data 2-35
- subsystem IDs 2-63
- swap RDF devices 2-28
- Symmetrix attributes, setting 2-50

Symmetrix device locks 2-3
Symmetrix model numbers xii

T
technical support xiii

U
Updating the host A-1

V
VCM devices 1-4
Virtual devices 1-9

