



EMC Solutions Enabler Symmetrix Array Management CLI

Version 6.3

PRODUCT GUIDE

P/N 300-002-939

REV A03

EMC Corporation

Corporate Headquarters:
Hopkinton, MA 01748-9103

1-508-435-1000

www.EMC.com

Copyright © 2005, 2006 EMC Corporation. All rights reserved.

Published September, 2006

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

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

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

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Preface	xv
PART 1	Concepts and Procedures
Chapter 1	Introduction
SYMCLI introduction	1-2
Symmetrix component overview.....	1-3
Directors in the Enginuity environment.....	1-3
Cache.....	1-4
Storage devices.....	1-4
SYMCLI usage tips.....	1-5
Obtaining SYMCLI help.....	1-5
Referencing devices and objects	1-6
Altering the command line response	1-7
Presetting the names and IDs.....	1-7
Time-saving command input tips	1-7
Truncated output	1-8
Performing operations on a device list	1-8
Chapter 2	Discovery
Discovering configuration and status data	2-2
Performing discovery	2-2
CLARiiON assisted discovery	2-3
Is the database file synchronized?	2-4
Scanning for new devices	2-4
Connectivity authorization.....	2-5
Synchronizing configuration updates with the database.....	2-6
Sync operations	2-6
SYMAPI configuration database file	2-7
Database file location.....	2-7
Changing the current database file	2-7
Access modes.....	2-8
Command modes: Online and offline.....	2-8
Database configuration information	2-9

Chapter 3 Configuration Data

Configuration data: Layer by layer	3-2
SCSI inquiries.....	3-3
Example: SCSI device information	3-3
Example: Devices listed by array ID	3-3
Example: Devices without capacity	3-4
Example: Devices with WWN.....	3-4
Example: Colon formatted WWN	3-5
Example: Pdevfile format	3-5
HBA information.....	3-5
Storage array data	3-7
List of arrays	3-7
Symmetrix configuration data	3-8
Viewing application registrations with Symmetrix access	3-9
Hosts connections to Symmetrix arrays	3-10
Viewing storage capacities to hosts.....	3-11
Director configuration data	3-11
Taking RA directors offline	3-14
Taking front-end director ports offline.....	3-14
Symmetrix external locks.....	3-15
SYMAPI semaphores.....	3-17
LRU cache management group.....	3-17
Viewing network services.....	3-18
Viewing memory board information.....	3-18
Mainframe CU image	3-19
Environmental data	3-19
Symmetrix devices	3-22
Device emulation	3-22
Device type overview	3-23
List of devices	3-24
Device service states	3-28
Device emulation type.....	3-29
Filtering device data with sympd.....	3-29
Filtering Symmetrix device data	3-30
Using the -all option	3-30
Detailed device configuration data	3-32
Releasing device external locks.....	3-32
CLARiiON devices.....	3-33
Example: CLARiiON devices	3-33
Disk-level data	3-34
List and show disks	3-34
Disk groups.....	3-35
Hotspares	3-36

Chapter 4 Host-based Access Control

Symmetrix Access Control operations overview	4-2
Access Control command	4-2
Access Control database	4-3
How to create or modify Access Control data	4-5
Command file execution	4-5
Minimum Access Control configuration	4-6
Initial ACL setup	4-6
How to create and manage access groups.....	4-7
Setting up an access group	4-7

User access IDs (PINs) for the AdminGrp	4-8
Editing and managing an access group	4-8
How to create and manage limited access to access pools	4-10
Setting up access pools	4-10
Editing and managing access pools	4-11
How to create and manage access control entries	4-13
Permissions defined	4-13
How to obtain access control information	4-18
How to release pending locked sessions	4-19
Grant/deny access strategies	4-20
Initially grant all permissions to everybody	4-20
Granting and denying default and legacy access	4-20
BCV setup strategy	4-22
TimeFinder/Snap setup strategy	4-22
SRDF setup strategy	4-22
How to back up and restore the access control database	4-23
Create a backup file	4-23
Restore the backup file to the access control database	4-23

Chapter 5 User Authorization

User Authorization operations overview	5-2
Roles	5-2
Users	5-4
User Authorization management	5-5
Enable and disable User Authorization	5-5
Set enforcement	5-5
Set server policy	5-6
Modify user to Role mappings	5-6
Command file usage	5-7
User Authorization monitoring	5-8
View User Authorization policies	5-8
View user-to-Role mappings	5-8
View Roles supported	5-8

Chapter 6 Grouping Devices

Overview	6-2
Grouping types	6-2
Group name services (GNS)	6-2
Group names and device names	6-3
Device group types	6-3
How to create a device group	6-5
Creating an empty device group	6-5
Device restrictions	6-5
Adding devices to a device group	6-6
Add devices by physical device name	6-6
Adding virtual devices	6-6
Adding devices to the target list	6-7
Adding all ungrouped devices	6-8
Listing device groups	6-9
Show details about a device group	6-9
Device groups: Exporting and importing device lists	6-11
Exporting a device list	6-11
Exporting and deleting a device group	6-11

Importing device lists	6-12
Renaming, moving, and deleting device groups	6-13
Renaming	6-13
Moving devices	6-13
Copy devices from a DG to a CG	6-14
Removing devices	6-15
Deleting a device group	6-15
Composite groups	6-16
What is an RDF consistency group?	6-16
Multi session consistency (MSC)	6-17
PowerPath consistency	6-17
Creating composite groups	6-18
Moving a composite group to or from MSC or PowerPath	6-20
Deleting a composite group	6-21
Adding standard devices	6-21
Removing standard devices	6-22
Renaming a composite group	6-23
Setting RDF group attributes	6-23
Composite group creation and output example	6-24
Composite group lists	6-25
Changing the state of devices in a composite group	6-27
SYMAPI optional behavior parameters for composite groups	6-27

Chapter 7 Group Name Services

Group name services overview	7-2
How the GNS daemon works	7-2
Shared group definitions with GNS	7-2
How GNS handles updates to group definitions	7-4
GNS and consistency groups	7-5
GNS device groups and SRDF	7-5
GNS behavior in client/server mode	7-6
Active vs. inactive group lists	7-6
GNS configuration	7-7
How to set up the GNS option	7-7
How to configure GNS in a multi-host environment	7-7
GNS user authentication	7-7
How to configure GNS for consistency	7-8
GNS daemon control	7-9
Enabling the watchdog	7-9
Managing the GNS daemon	7-9
GNS Daemon and Symmetrix external locks	7-10
GNS daemon log file	7-10
GNS daemon options file	7-10
How to mirror remote device group definitions	7-13
Controlling the mirrored group	7-13
Impact of remotely mirroring group definitions	7-13
BCV and remote BCV device alias name swap	7-14
Composite group logical device name support	7-15
Identify GNS groups	7-16
Group output	7-16
GNS backup and restore mechanism	7-17
Automatic GNS backups	7-17
Manual backups and restores via storgnsd	7-17
Troubleshooting GNS	7-18

Group name collisions.....	7-18
Inconsistent groups.....	7-18

Chapter 8 Statistics

Overview	8-2
About the symstat command	8-2
Reporting objects.....	8-2
Performance type options.....	8-3
REQUESTS statistics	8-4
Array statistics.....	8-4
Device statistics	8-5
SRDF/A session statistics	8-5
R-Copy session statistics	8-8
BACKEND statistics and PREFETCH activity.....	8-9
Returning BACKEND statistics	8-9
PREFETCH activity	8-9
Director statistics	8-10
Director I/O requests	8-10
R-Copy session statistics	8-11
PORT statistics.....	8-12
How to return director port statistics	8-12
R-Copy session director port statistics.....	8-12
MEMIO statistics	8-13
Example: Memory and I/O statistics	8-13
DMSP statistics	8-14
Device dynamic mirroring service policy statistics	8-14
Disk statistics	8-15
Specifying a disk	8-15
CACHE statistics	8-16
Front-end director statistics.....	8-16
LRU cache management statistics	8-16
SRDF/A session statistics	8-17
PATH statistics.....	8-20
CYCLE statistics	8-22
Example: All device groups.....	8-22

Chapter 9 Change Tracker

Introduction to change tracker	9-2
Practical uses.....	9-2
Change Tracker command summary.....	9-3
Monitoring data objects for changes	9-4
Creating a Change Tracker session	9-4
Marking a data object	9-5
Listing marked objects	9-6
Removing marked objects	9-6
Deleting marked objects.....	9-7
Viewing marked objects.....	9-8
Generating reports of changes	9-9
SUM reports.....	9-9
DELTA reports	9-9
Viewing reports	9-10
Report options	9-10
Example procedure.....	9-11
Using Change Tracker for RDF capacity planning.....	9-13

Chapter 10	Gatekeeper Devices	
	Overview	10-2
	Gatekeeper candidates	10-2
	How SYMCLI uses gatekeepers	10-3
	Defined gatekeepers	10-3
	How to size a gatekeeper	10-4
	Database considerations.....	10-4
	SYMCLI gatekeeper management	10-5
	Defining a gatekeeper.....	10-5
	Undefine a gatekeeper.....	10-5
	Listing the gatekeeper devices	10-6
	Associating with a device group	10-6
	Disassociating from a device group	10-6
Chapter 11	Events and Logs	
	Symmetrix event logs	11-2
	How to configure log options	11-2
	Monitoring Symmetrix events via SYMCLI.....	11-4
	Common audit log.....	11-6
	Event daemon and SNMP traps	11-8
Chapter 12	XML Structured Output	
	XML structured output data.....	12-2
	XSLT: XML data transformations	12-2
	XML output options	12-2
	Using XML Mode.....	12-3
PART 2	Operational Examples	
Chapter 13	Symmetrix Configuration Information Examples	
	Example 1: Verifying SYMAPI database configuration information.....	13-2
	Example 2: Obtaining more Symmetrix information.....	13-4
	Example 3: Obtaining more device information.....	13-7
	Example 4: Obtaining information about dynamic RDF devices	13-11
	Example 5: Obtaining information about virtual devices.....	13-12
	Example 6: Obtaining more device group information.....	13-15
	Example 7: Obtaining disk and audit information.....	13-21
	Example 8: Obtaining display output in XML format.....	13-26
	Example 9: Obtaining status information about GNS	13-28
Chapter 14	Access Control Examples	
	Example 1: Access Control group, pool, and ACE creation.....	14-2
	Example 2: Access Control and TimeFinder	14-6
	Example 3: Access Control and SRDF.....	14-7
	Example 4: Access Control and configuration manager	14-12
	Example 5: Access Control and older EMC software products.....	14-22
	Example 6: Access Control and EMC ControlCenter	14-24

Chapter 15 GNS Examples

Example 1: Importing device groups into GNS.....	15-2
Example 2: Using GNS to propagate a composite group definition	15-6

Chapter 16 Change Tracker Examples

Change Tracker example.....	16-2
-----------------------------	------

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

1-1	SYMCLI host/array interaction	1-2
1-2	Symmetrix components	1-3
4-1	ACLs and ACEs in the Symmetrix Access Control database	4-3
4-2	A Symmetrix site using Access Control	4-4
4-3	Creating access pools	4-10
4-4	Removing devices from a pool	4-11
4-5	Example default configuration	4-20
6-1	RDF consistency groups	6-18
7-1	GNS daemon controls GNS activity	7-2
7-2	Host-visible GNS state	7-3
7-3	GNS across multiple hosts	7-4
7-4	Mirrored Remote Group Definition	7-14
9-1	A DeltaMark bitmap table	9-2
9-2	SRDF configuration	9-13

1-1	SYMCLI help options	1-5
1-2	Command naming conventions	1-6
2-1	Database synchronization options	2-6
3-1	Device types	3-23
4-1	Initial access groups set	4-6
4-2	Initial delivered ACLs setup	4-6
4-3	Permissions: AccessType	4-13
5-1	Roles defined by permission	5-2
7-1	GNS daemon optional behavior parameters	7-11
8-1	Performance display options	8-3
9-1	Change Tracker command summary	9-3
9-2	SYMCLI report options	9-10
11-1	Log file configuration options	11-2
14-1	Example: Required access rights	14-12

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

Audience This manual is intended for command-line users and script programmers interested in details about managing arrays via SYMCLI commands.

Organization This manual describes how to manage Symmetrix arrays and their devices for an enterprise storage environment using the Base Management SYMCLI commands of the EMC Solutions Enabler software. The following defines the structure of this manual:

[Chapter 1, "Introduction,"](#) provides an introduction to the Solutions Enabler Symmetrix Command Line Interface (SYMCLI) used to manage your storage environment. The chapter describes terminology and the interface between the host and the Symmetrix systems.

[Chapter 2, "Discovery,"](#) describes the SYMCLI discovery process including details about the Symmetrix configuration database.

[Chapter 3, "Configuration Data,"](#) describes the how to use the SYMCLI to collect and display configuration data about arrays, Symmetrix devices, and disks.

[Chapter 4, "Host-based Access Control,"](#) describes how to set up access control mechanisms and how to perform Symmetrix access control actions using SYMCLI.

[Chapter 5, "User Authorization,"](#) describes how to set up user authorization and manage Roles using SYMCLI.

[Chapter 6, "Grouping Devices,"](#) describes the benefits of grouping devices, the types of groups, and how to create and modify them.

[Chapter 7, "Group Name Services,"](#) describes the optional Group Name Services (GNS) feature.

[Chapter 8, "Statistics,"](#) describes how to use SYMCLI to retrieve the performance-related statistics such as the number of I/O requests and throughput activities for the Symmetrix devices.

[Chapter 9, "Change Tracker,"](#) describes how to monitor Symmetrix data objects for change within your storage environment.

[Chapter 10, "Gatekeeper Devices,"](#) describes the function of gatekeepers and how to manage them in your system.

Chapter 11, “Events and Logs,” describes how to configure, manage, and query Symmetrix events and logs.

Chapter 12, “XML Structured Output,” describes the XML output option of the SYMCLI.

Chapter 13, “Symmetrix Configuration Information Examples,” provides real-world examples of how to use obtain various Symmetrix configuration information.

Chapter 14, “Access Control Examples,” provides real-world examples of using Access Control.

Chapter 15, “GNS Examples,” provides real-world examples of how to use the optional GNS functionality.

Chapter 16, “Change Tracker Examples,” provides examples of the SYMCLI actions and specific commands, which are used to measure and track changes to data on a group of Symmetrix volumes.

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

Related Documentation

Other Symmetrix publications of related interest are:

- ◆ *EMC Solutions Enabler Symmetrix CLI Command Reference*, EMC Corporation
- ◆ *EMC Solutions Enabler Installation Guide*, EMC Corporation
- ◆ *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide*, EMC Corporation
- ◆ EMC Host Connectivity Guides
- ◆ For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at <http://elabnavigator.emc.com>

Conventions Used in This Guide

EMC uses the following conventions for notes, cautions, warnings, and danger notices.

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



CAUTION

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

The following naming conventions are used in this manual:

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

In this manual, every occurrence of the word MVS in text or in symbolic syntax means OS/390 and z/OS.

In this manual, every occurrence of the word OSF1 in text or in symbolic syntax means Tru64 UNIX.

Typographical Conventions

This guide uses the following type style conventions:

- | | | |
|-------------------------|---|---|
| Palatino, bold | ◆ | Boldface text provides extra emphasis and emphasizes warnings, and specifies window names and menu items in text. |
| <i>Palatino, italic</i> | ◆ | New terms or unique word usage in text |
| | ◆ | Applies emphasis in examples and in references to book titles and sections |
| <i>Courier, italic</i> | ◆ | Identifies variables in a software syntax (non-literal notation) |
| <i>Courier</i> | ◆ | A fixed space font identifies files and pathnames, and is used in command line entries, displayed text, or program listings. |
| | ◆ | System prompts and displays and specific filenames or complete paths. For example:
working root directory [/user/emc]:

c:\Program Files\EMC\Symapi\db |
| Courier, bold | ◆ | Actual user entry in examples. For example:
symcfg discover |

Where to get help EMC support, product, and licensing information can be obtained as follows.

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

<http://Powerlink.EMC.com>

Technical support — For technical support, go to EMC WebSupport on Powerlink. To open a case on EMC WebSupport, you must be a WebSupport customer. Information about your site configuration and the circumstances under which the problem occurred is required.

Your comments Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this guide to:

techpub_comments@EMC.com



The Concepts and Procedures part of this product guide provides detailed descriptions of the management functionality and how to perform various management operations on arrays, devices, and disks using the SYMCLI commands of the EMC Solutions Enabler software. These concepts and procedures are described in the subsequent chapters as follows:

[Chapter 1, “Introduction,”](#) provides an introduction to the Solutions Enabler Symmetrix Command Line Interface (SYMCLI) used to manage your storage environment. The chapter describes terminology and the interface between the host and the Symmetrix systems.

[Chapter 2, “Discovery,”](#) describes the SYMCLI discovery process including details about the Symmetrix configuration database.

[Chapter 3, “Configuration Data,”](#) describes the how to use the SYMCLI to collect and display configuration data about arrays, Symmetrix devices, and disks.

[Chapter 4, “Host-based Access Control,”](#) describes how to set up access control mechanisms and how to perform Symmetrix Access Control actions using SYMCLI.

[Chapter 5, “User Authorization,”](#) describes how to set up user authorization and manage Roles using SYMCLI.

[Chapter 6, “Grouping Devices,”](#) describes the benefits of grouping devices, the types of groups, and how to create and modify them.

[Chapter 7, “Group Name Services,”](#) describes the optional Group Name Services (GNS) feature.

[Chapter 8, “Statistics,”](#) describes how to use SYMCLI to retrieve the performance-related statistics such as the number of I/O requests and throughput activities for the Symmetrix devices.

[Chapter 9, “Change Tracker,”](#) describes how to monitor Symmetrix data objects for change within your storage environment.

[Chapter 10, “Gatekeeper Devices,”](#) describes the function of gatekeepers and how to manage them in your system.

[Chapter 11, “Events and Logs,”](#) describes how to configure, manage, and query Symmetrix events and logs.

[Chapter 12, “XML Structured Output,”](#) describes the XML output option of the SYMCLI.

This chapter provides an introduction to the Solutions Enabler Symmetrix command line interface (SYMCLI) used to manage your storage environment. The chapter describes terminology and the interface between the host and the Symmetrix systems.

- ◆ [SYMCLI introduction.....](#)1-2
- ◆ [Symmetrix component overview](#)1-3
- ◆ [SYMCLI usage tips](#)1-5

SYMCLI introduction

The EMC[®] Solutions Enabler SYMCLI is a specialized library consisting of commands that can be invoked via the command line, or within scripts. These commands can be used to monitor device configuration and status, and perform control operations on devices and data objects within your managed storage complex. The target storage environments are typically Symmetrix[®]-based though some features are supported for CLARiiON[®] arrays as well.

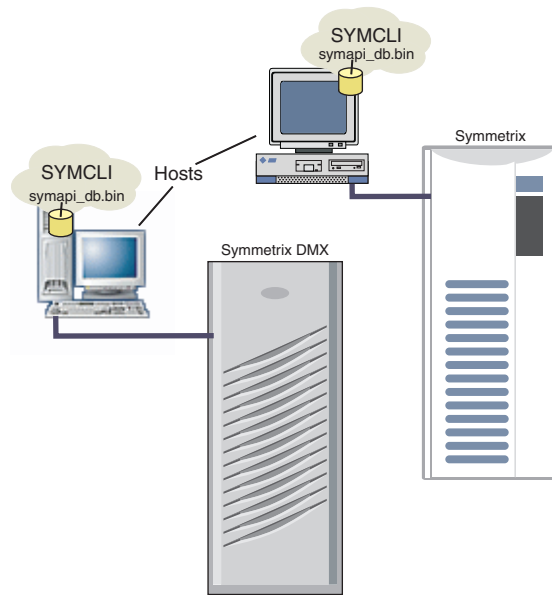


Figure 1-1 SYMCLI host/array interaction

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

To reduce the number of inquiries from the host to the storage arrays, configuration and status information is maintained in a Symmetrix host database file (called the Symmetrix configuration database; `symapi_db.bin` by default).

Symmetrix component overview

A Symmetrix system is an integrated cached disk array (ICDA) designed for online data storage. A host system and the Symmetrix array typically communicate via Fibre Channel or iSCSI interfaces (some models use a SCSI bus interface). Regardless of the hardware connection type, the communication protocol still recognizes the standard form of low-level SCSI commands.

Figure 1-2 illustrates the major functional components of a Symmetrix array and its connections to host systems.

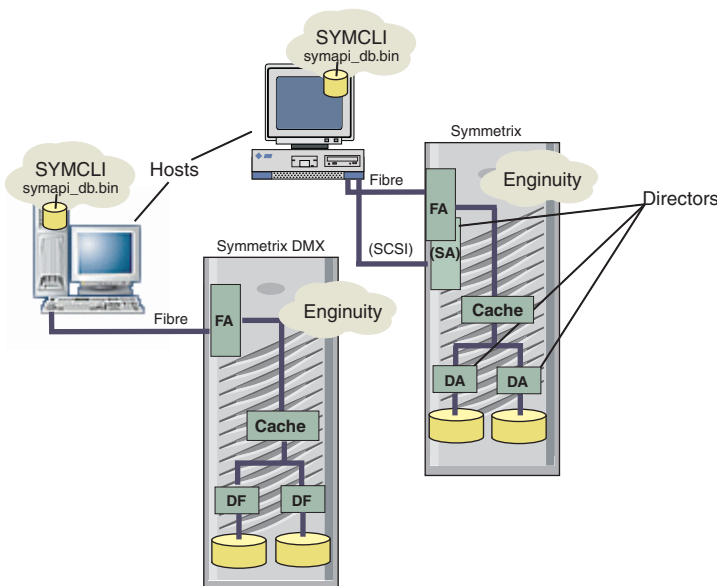


Figure 1-2 Symmetrix components

Directors in the Enginuity environment

In a Symmetrix array, a *front-end director* is usually a *Fibre Channel adapter* (FA) card that occupies a slot on the Symmetrix backplane or the DMX midplane. The FA card in DMX and older Symmetrix models can interface to a host via a fibre network. In older Symmetrix models, the host to Symmetrix interface connection could also be a SCSI bus that requires a SCSI adapter (SA) type front-end director. The front-end director can also be an ESCON director (EA), GIGE director (SE), or FICON (Fibre-ESCON) directors (FE).

A fibre-connected *disk director* (DF) or (DA) provide the back-end interface to disk arrays. The DF or DA is a card that occupies a slot on the Symmetrix backplane or DMX midplane. The processor on these director cards run the Symmetrix Enginuity™ operating system, which manages access to specific disk drives. These back-end directors transfer data from disk-to-cache and destage the write-bound data from cache-to-disk.

Another processor on the front-end director also runs the Symmetrix Enginuity operating system that handles I/O requests from the host. Front-end directors and disk directors share the cache. On a write request, the front-end director writes data to cache. A disk director later destages this data from cache to the appropriate physical disk.

Cache

Cache memory resides on cards that occupy slots on a Symmetrix backplane. Cache memory buffers I/O transfers between the director channels and the storage devices. Cache memory is segmented into *slots*, which are allocated to specific *tracks* in your storage devices. Cache access can be managed via *LRU* (Least Recently Used) ring partitions. Multiple LRUs can be established on a specific Symmetrix array and allocated to a percentage of cache (LRUs do not need to be of equal size). This enables a specific device pool size and cache partition to be allocated to each LRU based upon prioritized business needs.

Storage devices

From the perspective of software running on a host system, a Symmetrix array appears to be a number of *physical devices* connected to one or more I/O controllers. A host application addresses each of these devices using a physical device name. Each physical device defined in the configuration database has a specific set of attributes (such as vendor ID, product ID, revision level, and serial ID).

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 is mapped is called a *hyper-volume* or a *hyper*. A Symmetrix device may map to multiple hypers (containing identical copies of data) depending on its mirror configuration.

SYMCLI usage tips

This section describes some tips on obtaining help and streamlining your command line entries.

Obtaining SYMCLI help

SYMCLI provides multiple avenues to obtain help using the command set.

SYMCLI help command

Table 1-1 describes the various help options of the `symcli` command:

Table 1-1 SYMCLI help options

Command	Description
<code>symcli</code>	Provides the version number of the installed Symmetrix command line interface.
<code>symcli -h</code>	Describes how to use the SYMCLI command.
<code>symcli -v</code>	Displays the SYMCLI commands and a brief description of each command.
<code>symcli -env</code>	Displays a list of the environment variables that can be set for a SYMCLI session.
<code>symcli -def</code>	Displays a list of the environment variables that are set for the current SYMCLI session.

Command help

Command line help is available for each SYMCLI command. For example, to display command line help for the `symcli` command, enter:

```
symcli -h
```

In addition, each command has its own viewable man page for quick online reference purposes.

For example on UNIX, to view the man page for `symcli` command, enter:

```
man symcli
```

Note: In a UNIX environment, you need to have the SYMCLI man page directory (`/usr/symcli/man/`) included in your MANPATH environment variable.

On Windows, the man pages are located, by default, in the following directory:

```
C:\Program Files\EMC\symcli\man
```

These man pages can be viewed in any text editor. Clicking on a filename in this directory while using the Windows Explorer, you are prompted to select what program you want to use to open this file. Selecting Notepad will allow the text to be viewed or printed. If you use File Manager, the menu option **File Associate** will allow Notepad to be associated with files of this type.

Referencing devices and objects

There are a number of different terms to identify and specify a Symmetrix device using the SYMCLI commands:

PdevName or *pd* — Indicates a physical (host) device name

SymDevName or *dev* — Indicates a Symmetrix device name

LdevName or *ld* — Indicates a logical device name

Also, other Symmetrix objects require a specific notation to identify them in the command line. [Table 1-2](#) lists and describes the various SYMCLI command line notations and conventions to identify objects.

Table 1-2 Command naming conventions

Parameter/Ref.	Description	Examples
<i>CgName</i> (cg)	Composite group name	mycg1
<i>DbName</i> (db)	Relational database name	HR
<i>DgName</i> (dg)	Device group name	prod_1
<i>FileName</i>	Relational database file name	EMP_HIST.dbf
<i>LdevName</i> (ld)	Logical device name, either given at the command line or assigned automatically when a device is added to a device group	DEV001 BCV001
<i>LvolName</i> (lv)	The Logical Volume Manager's logical volume name	logvol1
<i>PdevName</i> (pd)	Physical device name	/dev/rdisk/c2t0d0s2 ^a
<i>SchemaName</i>	Relational database schema name	SCOTT
<i>SegmentName</i>	Relational database segment name	EMP_SEG
<i>SymDevName</i> (dev)	Symmetrix device name, unique for each Symmetrix array	001C
<i>SymmID</i> (sid)	Symmetrix array identifier number	010012392173
<i>TableName</i>	Relational database table name	EMP
<i>TblSpName</i>	Relational database tablespace name	tbl_space1
<i>VgName</i>	The Logical Volume Manager's volume group name	vol1

a. For Windows, the physical device name is of the form: \\.\physicaldevice2

Altering the command line response

You can alter the command line display behavior of your SYMCLI session.

Note: The UNIX command examples shown in this section are C shell based.

To always display the output of command responses in detail, set the verbose mode `-v` option as the default display behavior as follows:

```
setenv SYMCLI_VERBOSE 1
```

To suppress the verification prompts and streamline command operations, set the `-noprompt` option as the default display behavior as follows:

```
setenv SYMCLI_NOPROMPT 1
```

You can turn off or remove any of these environment variables. For example, to turn off the verbose mode, enter:

```
unsetenv SYMCLI_VERBOSE
```

Presetting the names and IDs

To reduce repeated key strokes for a group of commands that require the same argument, you can preset the device group name or the Symmetrix ID to a specified default value. The following variables can be preset to a specified value for the same argument of a command set:

`SYMCLI_DG` — defines the device group name for all `-g` arguments.

`SYMCLI_SID` — defines the Symmetrix ID for all `-sid` arguments.

For example, to set the Symmetrix ID to 100200000567 for the next group of commands in a sequence, enter:

```
setenv SYMCLI_SID 100200000567
```

Time-saving command input tips

The following are examples of some time-saving tips:

- ◆ Actions can be shortened by typing three or more unique characters as follows:

```
symcfg discover
```

or:

```
symcfg dis
```

- ◆ The last two or more digits of the Symmetrix Identifier (SID) can be used in place of the entire 12-digit SID as follows:

```
symcfg -sid 000002304324 sync
```

or:

```
symcfg -sid 24 sync
```

- ◆ Actions, keywords, and options are case-insensitive as follows:

```
symcfg LIST
```

or:

```
symcfg list
```

Physical device names can be abbreviated as follows:

```
sympd show /dev/rdisk/c2t1d1s2
```

or:

```
sympd show c2t1d1s2
```

Truncated output

In some cases, the output data from SYMCLI list commands may exceed the available column width. In these cases the returned data is truncated and appended with an asterisk (*). To view the full data, use the `show` command or verbose option, when available.

Performing operations on a device list

With one command, commands such as `symls`, `symmir`, and `symrdf` can perform control operations on a device group basis, individual device basis, or list basis. For device lists, you can either concatenate the list of device names (*LdevNames*) within the command line, or on UNIX only, specify a specific import file that contains the device names. For example, to perform a split operation on multiple devices listed in a file, use the form:

```
symrdf -g DgName split `cat LdevListFile`
```

The file, read by this UNIX command, must have logical device names with one *LdevName* per line (for `symmir`, one pair per line for establish or restore actions only).

For commands such as `symdev` and `symls`, for certain arguments you can specify a range of consecutive Symmetrix device names such as `-RANGE 0017:001C`, which would address 6 devices.

This chapter describes the SYMCLI discovery process.

- ◆ Discovering configuration and status data2-2
- ◆ Synchronizing configuration updates with the database2-6
- ◆ SYMAPI configuration database file.....2-7

Discovering configuration and status data

This chapter describes the process for retrieving array and device-level configuration and status information from the Symmetrix or CLARiiON environment known as *discovery*. Discovered configuration and status data for all arrays, as well their directors and devices, is maintained in a configuration database file on each host. Once your environment is discovered, you can direct information requests to retrieve array-level (high-level) data; or device-level (low-level) information from the storage environment using SYMCLI commands.

Performing discovery

During your first command line session, or if a configuration change has occurred, the configuration database must be built, or rebuilt, with the most complete and current information for all physical devices connected to your host. The following are the discovery syntax options:

```
symcfg  
  
discover [-all | -symmetrix | -clariion |  
         -pdev [-sid SymmID] | -sid SymmID]
```

To scan the hardware and rebuild the database, enter:

```
symcfg discover
```

This command scans all SCSI buses, collects information about all the arrays and devices found, and rebuilds the database with the collected device information and parameters from all local and remotely attached devices. You can limit the discovery to just Symmetrix arrays by specifying the `-symmetrix` option, just locally attached CLARiiON arrays by specifying the `-clariion` option, only physical device information by specifying the `-pdev` option, or just a specific Symmetrix array by specifying the `-sid` option (this option can also be used with the `-pdev` option to return only physical device information for a specific array). For more information about the Symmetrix configuration database, refer to [“SYMAPI configuration database file” on page 2-7](#).

CLARiiON assisted discovery

To discover CLARiiON arrays, Solutions Enabler requires authorization to access the targeted arrays and the hostname or IP address of the storage processors — SPA and SPB.

To discover all CLARiiON arrays, local or remote, you first need to create a connectivity authorization for the targeted arrays. For more information, refer to [“Connectivity authorization” on page 2-5](#).

Once you have established authorization via the `symcfg authorization` command, you can use the CLARiiON assisted discovery command to discover both local and remote CLARiiON arrays using the following command line:

```
symcfg discover -clariion -file AsstDiscoFile
```

where *AsstDiscoFile* is the fully-qualified pathname of a user created file containing the host name or IP address for each SPA and SPB of each targeted CLARiiON array, one array per line, as shown in the following example:

```
losat246      losat247  
EngDev101    EngDev112
```

Note: `symcfg discover` scans all SCSI buses on the host, not just those connected to Symmetrix arrays. This can take a significant amount of time to complete (refer to the post-installation section of the *EMC Solutions Enabler Installation Guide* for information on configuring Base Daemon caching options to improve performance). Only use `symcfg discover` if you have added or removed devices seen by the host. Also, if you had previously run `discover` and had subsequently removed one or more array(s), a later execution of `discover` will not remove information from the database relating to the removed Symmetrix array(s).

Is the database file synchronized?

Use the `symcfg verify` command to determine if the host configuration database file is synchronized with the current configuration of a specific Symmetrix array. If they are out of sync, the verify action returns code 24 (`CLI_C_NOT_IN_SYNC`). If they are in sync, the verify action returns code zero (the `CLI_C_SUCCESS` value) as shown below using the `echo $status` command.

```
symcfg verify -sid 814
```

```
The Symmetrix configuration and the database file are in sync.
```

```
echo $status
0
```

In addition, the `symcfg list -status` command provides a list of all arrays connected to your host, whether the configuration has changed, and if the array was discovered during the last discovery operation. The following is an example output:

```
symcfg list -status
```

```

                                S Y M M E T R I X
                                Mcode
SymmID      Attachment  Model   Version  Config Changed  Discovered
000187720758 Local      2000P-M2  5671     No              Yes
000190100097 Local      DMX3-24   5771     No              Yes
000000006206 Remote     2000P-M2  5671     Yes            Yes
000187900035 Remote     800-M2    5671     No              Yes
000187900041 Remote     800-M2    5671     Yes            Yes
000190300174 Remote     DMX3-6    5771     Yes            Yes
000190300175 Remote     DMX3-6    5771     No              Yes
000190300184 Remote     DMX3-6    5771     No              Yes

```

Scanning for new devices

When new devices are mapped to a host on nodes that have not yet been created, those nodes must exist before discover can find the devices along their path. A `symcfg scan` command searches your environment for devices accessible to a given, creating the nodes for the new or changes devices, and activating the necessary processing on the host system to recreate a list of accessible devices. It should be followed by a `symcfg discover -symmetrix`, if the device changes are associated with Symmetrix devices.

Connectivity authorization

Some arrays, such as CLARiiON arrays, require authorization information to provide to access the array. The `symcfg authorization` command can be used to supply this information for use in subsequent discovery operations. The `symcfg authorization` command allows you to list, add, update, or delete this connectivity information. The `update` action allows you to update the password of an existing entry. Use the following syntax:

```
symcfg
  authorization list

  authorization <add | update>
    -host HostName -username UserName -password PassWord

  authorization delete -host HostName -username UserName
```

Synchronizing configuration updates with the database

Once a discovery operation has occurred, configuration data in the host configuration database file can be synchronized with the storage environment, which is less performance-intensive than a full discover operation.

A *sync* operation interrogates just the known arrays (previously discovered) that are accessible from the host, and updates the configuration and status information in the configuration database file. In addition, this synchronization can be limited to a specific array or array type, where a configuration change may have occurred.

Sync operations

A sync operation only updates state, mode, and configuration information (if a configuration change has occurred). It does not update statistical information. Refer to [Chapter 8, "Statistics,"](#) for information about obtaining Symmetrix performance statistics.

The *sync* option updates data for all known entities in the configuration database file, but does not scan for SCSI devices on the host. That means that any newly configured physical devices will not be added to the database during a sync operation.

Synchronizing data is an array-specific operation. Both Symmetrix and CLARiiON arrays use different options to limit the scope to a specific data set.

Table 2-1 Database synchronization options

Array	Command and options	Description
Symmetrix	<pre>symcfg sync [-sid SymmID] [-rdf -bcv -local -dirsts -snap -cfgmgr -rcopy -env_data]</pre>	Performs a basic sync of all known Symmetrix entities. Limits the scope to the entities of a specified Symmetrix array. Options limit the scope to one of the following: - updates RDF information - updates BCV information - updates local Symmetrix array information - updates director status information - updates TimeFinder®/Snap information - updates disk space and Symmetrix configuration metrics - updates R-Copy information
CLARiiON	<pre>symcfg -clarion sync [-cid ClarID]</pre>	Performs a basic sync of all known CLARiiON entities. Limits the scope to the entities of a specified CLARiiON array.

SYMAPI configuration database file

A SYMAPI configuration database (.bin) file, which is stored on the host system, contains the physical configuration information of SCSI devices and Symmetrix parameters that define your entire storage complex. More than one configuration database file may be required to support your operational needs.

The SYMAPI configuration database is sometimes referred to as the host configuration database, SYMAPI database (because of how the file is named), or the Symmetrix database file. All of these names are referring to the same configuration database file, `symapi_db.bin`, described next.

Note: For information about discovering your storage environment and building your Symmetrix configuration database, refer to [“Discovering configuration and status data” on page 2-2](#).

Database file location

On UNIX, the default pathname for the configuration database file is:

```
/var/symapi/db/symapi_db.bin
```

On Windows, the default configuration database path is:

```
C:\Program Files\EMC\Symapi\db\symapi_db.bin
```

On OpenVMS, the default configuration database path is:

```
SYMAPI$DB:symapi_db.bin
```

You can create an additional configuration database (.bin) file to meet your requirements.

Note: In a multi-database environment, you should carefully verify that you are operating on the correct database. Always confirm the current database before applying a command. For safe command line environments, it is recommended that you use the common (default) database.

Changing the current database file

To check which Symmetrix configuration database file is being used, enter:

```
symcli -def
```

To change the current Symmetrix configuration database file, you need to modify the environment variable `SYMCLI_DB_FILE`.

For example, on UNIX, to change the database file to `symlbackup_db.bin`, enter:

```
setenv SYMCLI_DB_FILE /var/symapi/db/symlbackup_db.bin
```

To perform the same operation on Windows, enter:

```
set SYMCLI_DB_FILE=C:\Program Files\EMC\Symapi\db\symlbackup_db.bin
```

Access modes

The SYMCLI commands utilize different modes to access the Symmetrix configuration database file:

read/write — Commands that control and/or modify database parameters, read the database file into memory, and provide simultaneous modification of both the in-memory database and the database file. During this access cycle, the database file is locked.

read/no write — Commands that list or show database parameters, read the database file into memory and allow modifications to the in-memory database. No modifications to the database file occur. During the access cycle, the database file is not locked.

Command modes: Online and offline

SYMCLI commands can either run in online or offline mode. Commands that execute in online mode, such as configuration control operations, automatically attempt to gather the latest state and mode information from the arrays, and update the in-memory database and configuration database file on the host. In the event a configuration change has occurred, commands that execute in online mode will attempt to discover the changed entity to retrieve and load any updated configuration information.

Commands that can execute in offline mode, such as `symcfg list`, retrieve data exclusively from the configuration database.

Inhibiting database synchronization

You can force some commands to operate in offline mode using the environment variable `SYMCLI_OFFLINE`, which inhibits accessing the Symmetrix array to update the database.

For example, to globally force these commands to their offline mode (`-offline` option), enter:

```
setenv SYMCLI_OFFLINE 1
```

Note: You may need to refresh the database if you execute commands while offline mode is enabled or execute commands that normally run in the offline mode, such as `symcfg list`. Most display commands can be run in the offline mode, which provides the fastest response, and does not require access to the Symmetrix array.

Database configuration information

The `symcfg -db` command, returns basic configuration information about the current database in use. The following example may not match the values present in your configuration:

```
Type of SYMAPI Database                : Full
Host Node Name which discovered the DB  : HostNode
Host OS Type which discovered the DB    : SunOS

Version of SYMAPI Library which discovered the DB : 6.X.X.X (Edit Level: 595)
Version of SYMAPI Library which wrote the DB      : 6.X.X.X (Edit Level: 595)
Min Edit Level of SYMAPI Lib Req'd to Read the DB : 478

Time Database Was Last Synchronized      : Mon Nov 29 07:10:22 200X
Time Any Device Group Was Last Modified   : Tue Nov 16 15:17:09 200X

GNS State                                : Disabled
```

Note: The information provided includes the minimum edit level of the SYMAPI library required to read the database. That means that any SYMAPI library with this version and edit level or higher can read the current database.

This chapter describes the how to use the SYMCLI to collect and display configuration data about arrays, Symmetrix devices, and disks.

◆ Configuration data: Layer by layer	3-2
◆ SCSI inquiries	3-3
◆ Storage array data.....	3-7
◆ Symmetrix devices.....	3-22
◆ CLARiiON devices	3-33
◆ Disk-level data	3-34

Configuration data: Layer by layer

Solutions Enabler SYMCLI allows you to gather and display various levels of information about your storage environment. Like an onion, each layer of your storage environment reveals new, or slightly different, perspective on a given entity. For example, from the array perspective, information of a specific device is very limited. It would be included as part of the count of devices on the array and the devices addressing as mapped to a specific director could be obtained. Yet, by delving down to the device-level data—using SYMCLI to gather data from the device perspective—you can obtain much greater detail about the device itself, such as capacity, status, mirrors. From this device view, the array layer becomes obscured as it reflects the relationship from the device's perspective, such as the associated array ID.

By peeling back the many layers, you can gain a deeper understanding of your storage environment. SYMCLI provides various commands that can be used to focus your queries on the SCSI-, array-, device-, or disk-level.

- ◆ **SCSI-Level** — Data can be returned at the SCSI level, whereby the SYMCLI issues SCSI INQUIRY and SCSI READ CAPACITY to return low-level physical device data (such as vendor, configuration, and basic configuration) and host HBA information (such as vendor, model, firmware, and basic configuration).
- ◆ **Array-Level** — Array queries allow you to return further detailed data about the configuration of one or all arrays. The data returned includes host relationship information (local or remote), cache size, and number of devices. Further details include vendor specific configuration information, ports, flags, and adapters.
- ◆ **Device-Level** — As you go deeper into your storage array, you can gather more detailed information about the devices in a given array (or all arrays, either locally or remotely attached). Device-level data includes capacity, cache, emulation, configuration, group, and usage information that is critical to all other storage management operations.
- ◆ **Disk-Level** — Return detailed information about the raw disks from which your storage device are contrived, including detailed identification, vendor, capacity, and hyper information.

SCSI inquiries

The `syminq` command can be used to obtain SCSI disk device information via a SCSI INQUIRY command, and optionally SCSI READ CAPACITY command, on host HBAs and one or all locally attached physical devices.

You can limit the scope of an inquiry to Symmetrix, CLARiiON, StorageWorks, or HDS devices. When specifying a specific array, options can be specified to control what data is returned and its presentation as specified in the `syminq` manpage, which can be found in the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

In addition, `syminq` can be used to list all HBAs on the local host. Using the options available, the scope of this request can be limited to list only the Fibre HBAs, SCSI HBAs, iSCSI HBAs, or HBAs derived from a SNIA API query — in the UNIX environment, if the SNIA libraries are unavailable, HBA information is obtained from the lost log files.

Note: Data returned from issuing any `syminq` command is not stored in the configuration database.

The returned data and output format is determined by the scope of options specified on the command line and the array information available. The following examples illustrate a few typical option combinations and an example of the possible output.

Example: SCSI device information

The following is an example of a SCSI inquiry of all devices listed by physical device name:

syminq

Device		Product			Device	
Name	Type	Vendor	ID	Rev	Ser Num	Cap (KB)
/dev/sdv	BCV	EMC	SYMMETRIX	5771	1600028000	2112000
/dev/sdw	BCV	EMC	SYMMETRIX	5771	1600029000	2112000
/dev/sdx	BCV	EMC	SYMMETRIX	5771	160002A000	2112000
/dev/sdy	BCV	EMC	SYMMETRIX	5771	160002B000	2112000
/dev/sdz	BCV	EMC	SYMMETRIX	5771	160002C000	2112000
/dev/sdaa	BCV	EMC	SYMMETRIX	5771	160002D000	2112000

Example: Devices listed by array ID

The following is an example of a SCSI inquiry of Symmetrix devices listed by Symmetrix ID:

syminq -symmids

Device		Symmetrix			Device	
Name	Type	ID	Rev	Ser Num	Cap (KB)	
/dev/sdv	BCV	000190300016	5771	1600028000	2112000	
/dev/sdw	BCV	000190300016	5771	1600029000	2112000	
/dev/sdx	BCV	000190300016	5771	160002A000	2112000	
/dev/sdy	BCV	000190300016	5771	160002B000	2112000	
/dev/sdz	BCV	000190300016	5771	160002C000	2112000	
/dev/sdaa	BCV	000190300016	5771	160002D000	2112000	
/dev/sdab	BCV	000190300016	5771	160002E000	2112000	

The following is an example of a SCSI inquiry of CLARiiON devices listed by CLARiiON ID:

syminq -cids

Device		Clariion		Device	
Name		ID	Rev	Ser Num	Cap (KB)
/dev/rdisk/c5t0d0s2		APM00034501433	0216	09000017	2097152
/dev/rdisk/c5t0d1s2		APM00034501433	0216	07000017	2097152

Example: Devices without capacity

The following is an example of a SCSI inquiry of Symmetrix devices without issuing a SCSI READ CAPACITY:

syminq -symmids -nocap

Device		Symmetrix		Device	
Name	Type	ID	Rev	Ser Num	
/dev/sdv	BCV	000190300016	5771	1600028000	
/dev/sdw	BCV	000190300016	5771	1600029000	
/dev/sdx	BCV	000190300016	5771	160002A000	
/dev/sdy	BCV	000190300016	5771	160002B000	
/dev/sdz	BCV	000190300016	5771	160002C000	

The following is an example of a SCSI inquiry of CLARiiON devices without issuing a SCSI READ CAPACITY:

syminq -cids -nocap

Device		Clariion		Device	
Name		ID	Rev	Ser Num	
/dev/rdisk/c5t0d0s2		APM00034501433	0216	09000017	
/dev/rdisk/c5t0d1s2		APM00034501433	0216	07000017	

Example: Devices with WWN

The following is an example of a SCSI inquiry of Symmetrix devices returning the WWN for each device:

syminq -sym -wwn

Device			Device	
Name	Num	Array ID	WWN	
/dev/sdv	0028	000190300016	60060480000190300016533030303238	
/dev/sdw	0029	000190300016	60060480000190300016533030303239	
/dev/sdx	002A	000190300016	60060480000190300016533030303241	
/dev/sdy	002B	000190300016	60060480000190300016533030303242	
/dev/sdz	002C	000190300016	60060480000190300016533030303243	
/dev/sdaa	002D	000190300016	60060480000190300016533030303244	
/dev/sdab	002E	000190300016	60060480000190300016533030303245	

Example: Colon formatted WWN

The following is an example of how the `-colon` option can be used to return a list of devices with WWN for each device using colons as spacers:

```
syminq -sym -wwn -colon
```

Device			Device
-----			-----
Name	Num	Array ID	WWN
-----	-----	-----	-----
/dev/sdv :30:30:30:32:38	0028	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdw :30:30:30:32:39	0029	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdx :30:30:30:32:41	002A	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdy :30:30:30:32:42	002B	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdz :30:30:30:32:43	002C	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdaa :30:30:30:32:44	002D	000190300016	60:06:04:80:00:01:90:30:00:16:53
/dev/sdab	002E	000190300016	60:06:04:80:00:01:90:30:00:16:53

Example: Pdevfile format

The following is an example of a SCSI inquiry of Symmetrix devices returns a list of device names in a format for use as pdevfile:

```
syminq -pdevfile
```

```
# Symm_id    pdev                dev dir  dir_port
000187900771 /dev/rdisk/c2t0d0s2 0017 15D  0
000187900771 /dev/rdisk/c2t0d1s2 0018 15D  0
000187900771 /dev/rdisk/c2t0d2s2 0019 15D  0
000187900771 /dev/rdisk/c2t0d3s2 001A 15D  0
```

HBA information

To obtain a list of all of the local host's HBAs, use the `syminq hba` command entry.

You can filter the returned data by using one or a combination of available options, such as to list only the Fibre (`-fibre`), SCSI (`-scsi`), or iSCSI (`-iscsi`) HBAs.

In addition, the `-snia` option allows you to return HBA information via the native SNIA SMI-S Provider rather than by issuing a SCSI inquiry. The returned data will be exactly the same as the data returned via a `-fibre` SCSI inquiry; the only difference is how the data was obtained.

Example: HBA inquiries

To issue a SCSI INQUIRY and READ CAPACITY command on all Fibre HBAs, enter:

```
syminq hba -fibre
```

The following is an example of the output returned:

```
Host Name : hst115
```

```
HBA Type      : FibreChannel
HBA Name      : Emulex-LP8000-1
Vendor        : Emulex Corporation
```

```
Model          : LP8000
Serial Number   : 0000c9238053
Firmware Version : 3.20 (D2D3.20X3)
Driver Version  : 5.00f; HBAAPI v1.3, 3-29-02
Host WWN        : 20000000c9238053
Number of Ports : 1

    Port WWN      : 10000000c9238053
    Port name     : /devices/pci@1f,4000/fibre-channel@2
    Port type     : NLPort
    Port FCID     : 2493971
    Port speed    : 1gbit
    Supported speed : 1gbit
    Port state    : Online
```

The returned data includes:

- ◆ HBA information — HBA type, HBA name, Vendor, Model, Serial Number, Firmware Version, Driver Version, Host WWN (World Wide Name), and Number of Ports.
- ◆ Host Port Details — Port WWN, Port name, Port type, Port FCID (Fibre Channel ID), Port speed, Supported speed, and the Port state.

Note: HBA returned data will vary slightly based upon type.

Storage array data

To query your storage environment configuration data, you must first identify the arrays in your storage environment. Each array has a serial number that is used to uniquely identify it. On Symmetrix arrays, it is called a Symmetrix ID (SID) and on CLARiiON arrays a CLARiiON ID (CID).

A storage array contains a complex set of data. The `symcfg` command provides arguments and options to use to pinpoint specific components and access the information you need. The storage array data includes the following:

- ◆ **Storage Arrays** — A list of all arrays hatched to the host and details about that arrays configuration.
- ◆ **Registered Applications** — Applications registered with SYMAPI that have accessed all, or a specific, Symmetrix array(s) to which your host is connected.
- ◆ **Host Connections** — Detailed information about the hosts that have accessed a Symmetrix array the host node name and the Symmetrix ID, director/port information, and array capacity.
- ◆ **Director Information** — Configuration and status information about all directors of a given array including address, port, and status information. Details can be limited to a specific type of director.
- ◆ **SELs (Symmetrix External Locks) and Semaphores** — Status information on SELs (by number or type) and SYMAPI semaphores.
- ◆ **Cache Management** — Information on the LRU cache including the cache slots that each LRU occupies, and the percentage of the total cache utilized.
- ◆ **Network Services** — A list of network services available SYMAPI client applications.
- ◆ **Environment Data** — Detailed information on the memory boards, and the arrays environmental data, including fans and power supplies, can be obtained.
- ◆ **CU Images** — Information for mainframe users.

List of arrays

Using SYMCLI, you can obtain a list of all the accessible Symmetrix arrays by SID or CLARiiON arrays by CID, including the model number and the number of accessible devices.

For Symmetrix arrays, you can use SYMCLI commands to obtain detailed configuration information about a specific Symmetrix array. This high-level information (such as the SID) can be used with subsequent commands to obtain configuration information for each of the directors and devices within an array.

To obtain a list of all Symmetrix IDs for the arrays connected to your host, with one entry for each Symmetrix array accessible from the host, enter:

```
symcfg list
```

S Y M M E T R I X

SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000187720758	Local	2000P-M2	5671	51200	196	1507
000190100097	Local	DMX3-24	5771	98304	20	32088
000000006206	Remote	2000P-M2	5671	32768	0	4139
000187900035	Remote	800-M2	5671	8192	0	1314
000187900041	Remote	800-M2	5671	8192	0	960
000190300174	Remote	DMX3-6	5771	16384	0	335
000190300175	Remote	DMX3-6	5771	16384	0	914
000190300184	Remote	DMX3-6	5771	16384	0	417

To obtain a list of all CLARiiON IDs for the arrays connected to your host, with one entry for each CLARiiON array accessible from the host, enter:

```
symcfg list -clariion
```

Symmetrix configuration data

To obtain detailed configuration and status data for all the Symmetrix arrays, enter (the return data has been truncated):

```
symcfg list -v
```

```
<...>
```

```
Symmetrix ID: 000190300180 (Remote)
```

```
Time Zone : EDT
```

```

Product Model                : DMX3-6
Symmetrix ID                  : 000190300180

Microcode Version (Number)    : 5771 (168B0000)
Microcode Date                 : 06.22.2006

Microcode Patch Date          : 06.22.2006
Microcode Patch Level          : 86

Cache Size (Mirrored)         : 16384 (MB)
# of Available Cache Slots     : 460798
# of PermaCache Slots In Use   : 6
Max # of System Write Pending Slots : 368998
Max # of DA Write Pending Slots : 184498
Max # of Device Write Pending Slots : 18440

Symmetrix Total Operating Time : 168 days, 19:40:55
Symmetrix Power ON Time        : Wed Feb 15 13:59:16 2006
Symmetrix Last IPL Time (Cold) : Fri Feb 17 09:01:21 2006
Symmetrix Last Fast IPL Time (Hot) : Mon Jul 31 14:49:14 2006

Host DB Sync Time              : Thu Aug 3 09:36:56 2006
Symmetrix CLI (SYMCLI) Version : T6.3.0.7 (Edit Level: 762)
Built with SYMAPI Version      : T6.3.0.7 (Edit Level: 762)
SYMAPI Run Time Version        : T6.3.0.7 (Edit Level: 762)
SYMAPI Server Version          : unavailable

Number of Configured (Sym) Devices : 329
Number of Visible (Host) Devices   : 0
Number of Configured Actual Disks  : 59
Number of Configured Hot Spares    : 1
Number of Unconfigured Disks       : 0
Maximum number of hypers per disk  : 255

Number of Powerpath Devices        : 0
Powerpath Run Time Version         : N/A

SDDF Configuration State           : Enabled
Configuration Change State         : Enabled
WORM Configuration Level            : N/A
WORM Characteristics               : N/A

Symmetrix Configuration Checksum    : 10D2BB
Switched RDF Configuration State    : Enabled
Concurrent RDF Configuration State  : Enabled
Dynamic RDF Configuration State     : Enabled
Concurrent Dynamic RDF Configuration : Enabled
RDF Data Mobility Configuration State : Disabled
```

```

Access Control Configuration State : Disabled
Device Masking (VCM) Config State : Mixed
VCMdb Access Restricted State     : N/A
Multi LRU Device Assignment       : BY_NUMBER
Disk Group Assignments            : Not in Use
Hot Swap Policy                   : Permanent

Parity Raid Configuration         : N/A
Raid-5 Configuration              : RAID-5 (3+1)
PAV Mode                          : DynamicStandardPAV
PAV Alias Limit                   : 255

SRDF/A Maximum Host Throttle (Secs) : 0
SRDF/A Maximum Cache Usage (Percent) : 94

```

Viewing application registrations with Symmetrix access

You can view all the applications registered with SYMAPI that have accessed all, or a specific, Symmetrix array(s) to which your host is connected.

For example, to list all the applications that have worked with Symmetrix 039, enter:

```
symcfg list -applications -sid 039
```

```
Symmetrix ID      : 000187900039
```

Host		Application			
Node Name	IP Address	ID	Vendor ID	Version	Attr
Nodex5	172.23.192.115	SYMSTAR	EMC Corp	6.3.0.7	(R)
NodeXXXX	172.23.193.180	SYMRDF	EMC Corp	6.2.1.0	(R)
		storapid	EMC Corp	6.2.733.1	-
		SYMCFG	EMC Corp	6.2.1.0	-
Nodex82	172.23.193.182	SYMCFG	EMC Corp	6.3.0.86	(R)
		SYMCFG	EMC Corp	6.4.0.19	(R)
		SYMCFG	EMC Corp	6.4.0.18	(R)
		SYMCFG	EMC Corp	6.3.0.6	(R)
Nod79	172.23.192.79	storapid	EMC Corp	6.3.0.9	-
		SYMRDF	EMC Corp	6.3.0.2	-
		SYMSTAR	EMC Corp	6.3.0.7	-
		storapid	EMC Corp	6.3.0.7	-
		SYMCFG	EMC Corp	6.3.0.7	-
		SYMRDF	EMC Corp	6.3.0.7	-
NodeX50	172.23.161.69	storapid	EMC Corp	6.2.733.1	-
		OPTIMIZER	EMC Corp	5.2.0.10	-

Legend for Attribute(s):

(M) : Application Registered Remotely via RDF links (Multi-Hop).

(R) : Application Registered Remotely via RDF links (One-Hop).

Hosts connections to Symmetrix arrays

The `-connections` option of `symcfg list` command returns the host connections to array. Only hosts that have at least one registered application are listed. To view all the hosts that have accessed a Symmetrix array to which your host is connected, enter (omit the `-sid` value for all arrays):

```
symcfg list -connections -capacity -sid 097
```

Host		Symmetrix		Capacity (GB)		
Node Name	ID	Director	Port	Mapped STDs	Mapped BCVs	Paired BCVs
6I	000190100097	FA-3B	0	135.0	67.4	8.4
6I totals:				130.8	54.8	8.4
LBQA0074	000190100097	FA-3B	0	135.0	67.4	8.4
LBQA0074 totals:				0.0	0.0	0.0
api105	000190100097	FA-3A	0	134.9	0.0	0.0
api105 totals:				134.9	0.0	0.0
api150	000190100097	FA-3A	0	134.9	0.0	0.0
api150 totals:				134.9	0.0	0.0
api31	000190100097	FA-7A	0	404.6	0.0	0.0
api31 totals:				404.6	0.0	0.0
lbqa0074	000190100097	FA-3B	0	135.0	67.4	8.4
lbqa0074 totals:				0.0	0.0	0.0

In the example above, the display of the host connections are sorted according to each Symmetrix array. To list all the host connections, sorted by host names as shown below, of Symmetrix arrays to which your host is connected, enter:

```
symcfg list -connections -sorthost -sid 097
```

Host		Symmetrix				
Node Name	IP Address	OS Name	OS Revision	ID	Director	Port
6I	172.23.195.65	WinNT	5.2.3790	000190100097	FA-3B	0
LBQA0074	172.23.195.74	WinNT	5.2.3790	000190100097	FA-3B	0
api105	172.23.192.105	SunOS	5.8	000190100097	FA-3A	0
api150	172.23.192.150	WinNT	5.2.3790	000190100097	FA-3A	0
api31	172.23.192.31	SunOS	5.8	000190100097	FA-7A	0
lbqa0074	172.23.195.74	WinNT	5.2.3790	000190100097	FA-3B	0

Viewing storage capacities to hosts

The `-connections` option can be used in conjunction with the `-capacity` option to view the total Symmetrix storage space connected to each registered host based on devices mapped to each director and port.

For example, to list the allocated storage capacities of all the host names that have worked with Symmetrix 097, enter:

```
symcfg list -connections -capacity -sid 097
```

Host		Symmetrix		Capacity (GB)		
Node Name	ID	Director	Port	Mapped STDs	Mapped BCVs	Paired BCVs
6I	000190100097	FA-3B	0	135.0	67.4	8.4
6I totals:				130.8	54.8	8.4
LBQA0074	000190100097	FA-3B	0	135.0	67.4	8.4
LBQA0074 totals:				0.0	0.0	0.0
api105	000190100097	FA-3A	0	134.9	0.0	0.0
api105 totals:				134.9	0.0	0.0
api150	000190100097	FA-3A	0	134.9	0.0	0.0
api150 totals:				134.9	0.0	0.0
api31	000190100097	FA-7A	0	404.6	0.0	0.0
api31 totals:				404.6	0.0	0.0
lbqa0074	000190100097	FA-3B	0	135.0	67.4	8.4
lbqa0074 totals:				0.0	0.0	0.0

The data returned is already sorted by host, therefore the `-sorthost` option is not necessary and will return the same result.

Director configuration data

SYMCLI can be used to gather information about the Symmetrix directors. The following options are available to gather information about the various director types:

- DIR — All types of directors.
- CA — Channel directors.
- DA — Disk directors.
- EA — ESCON directors.
- SE — GIGE directors.
- EF — FICON (Fibre-ESCON) directors.
- RA — RDF directors.
- SA — Front-end (SCSI, GIGE, or Fibre) directors.
- FA — Front-end (Fibre only) directors.

To get configuration and status information about all directors on all attached arrays, enter:

```
symcfg -dir all list -sid 097
```

```
Symmetrix ID: 000190100097
```

```

      S Y M M E T R I X      D I R E C T O R S

      Ident  Symbolic  Numeric  Slot  Type           Status
      -----
      DF-1A   01A       1         1   DISK             Online
      DF-2A   02A       2         2   DISK             Online
      FA-3A   03A       3         3   FibreChannel     Online
      FA-4A   04A       4         4   FibreChannel     Online
      DF-5A   05A       5         5   DISK             Online
      DF-6A   06A       6         6   DISK             Online
      FA-7A   07A       7         7   FibreChannel     Online
      FA-10A  10A       10        10   FibreChannel     Online
      DF-11A  11A       11        11   DISK             Online
      DF-12A  12A       12        12   DISK
      < . . . >

```

This command returns the following information:

- ◆ The director ID
- ◆ The Symmetrix slot number in which the director resides
- ◆ The director type
- ◆ The director status, whether online or offline

Director information by type

You can get configuration and status information about one or all directors of a certain type. For example, to get information about all the front-end directors on all attached arrays (SA, SE, or FA), enter:

```
apil180.lss.emc.com~root>> symcfg -sa all list -sid 039
```

```
Symmetrix ID: 000187900039
```

```

      S Y M M E T R I X      D I R E C T O R S

      Ident  Symbolic  Numeric  Slot  Type           Status
      -----
      FA-2C   02C       34         2   FibreChannel     Online
      FA-15C  15C       47        15   FibreChannel     Online
      FA-16C  16C       48        16   FibreChannel     Online
      FA-2D   02D       50         2   FibreChannel     Online
      FA-15D  15D       63        15   FibreChannel     Online
      FA-16D  16D       64        16   FibreChannel     Online

```

Director port data

Use the `-port` option to refer to which ports are online or offline on SA, FA, SE, or RA directors. For example, to get the port status on all FA directors on all attached arrays, enter:

```
symcfg -fa all list -port -sid 097
```

Symmetrix ID: 000190100097

S Y M M E T R I X D I R E C T O R P O R T S

Ident	Type	Director Status	Port Status				Connection Status			
			P0	P1	P2	P3	P0	P1	P2	P3
FA-3A	FibreChannel	Online	ON	ON	N/A	N/A	X	-	-	-
FA-4A	FibreChannel	Online	ON	ON	N/A	N/A	-	-	-	-
FA-7A	FibreChannel	Online	ON	ON	N/A	N/A	X	-	-	-
FA-10A	FibreChannel	Online	ON	ON	N/A	N/A	-	-	-	-
FA-13A	FibreChannel	Online	ON	ON	N/A	N/A	-	-	-	-
FA-14A	FibreChannel	Online	ON	ON	N/A	N/A	-	-	-	-
FA-3B	FibreChannel	Online	ON	ON	N/A	N/A	X	-	-	-

< . . . >

This return output includes the director identification (number and type), director status, available ports and status, and connection status. In the connection status, the (X) indicates that a Fibre port is connected to a Fibre port (HBA, Switch, or RF Director) and the dash (-) indicates that the Fibre port is not connected.

Addresses of devices mapped to directors

Use the `-address` option to identify the address information for devices accessible through specific directors via a host-based view of the storage environment. For example, to list the address information for all director types of Symmetrix 097, enter:

```
symcfg list -dir all -address -sid 097
```

Symmetrix ID: 000190100097

Director				Device Name		Attr	Address	
Ident	Symbolic	Port	Sym	Physical		VBUS	TID	LUN
FA-3A	03A	0	0E2A	/dev/vx/rdmp/c5t0d0s2		0	00	000
			0E2B	/dev/vx/rdmp/c5t0d1s2		0	00	001
			04AA	/dev/vx/rdmp/c5t0d2s2	(M)	0	00	002
			04AE	/dev/vx/rdmp/c5t0d3s2	(M)	0	00	003
			04B2	/dev/vx/rdmp/c5t0d4s2	(M)	0	00	004
			04B6	/dev/vx/rdmp/c5t0d5s2	(M)	0	00	005
			04BA	/dev/vx/rdmp/c5t0d6s2	(M)	0	00	006
			04BE	/dev/vx/rdmp/c5t0d7s2	(M)	0	00	007
			04C2	/dev/vx/rdmp/c5t0d8s2	(M)	0	00	008
			04C6	/dev/vx/rdmp/c5t0d9s2	(M)	0	00	009
FA-7A	07A	0	0E2C	Not Visible		0	00	000
			0E2D	Not Visible		0	00	001
			046A	Not Visible		0	00	002
			046B	Not Visible		0	00	003
			046C	Not Visible		0	00	004

< . . . >

Next available device address

Used in conjunction with the `-address` option, the `-availability` option returns the next available address that can be used for a device. Those VBUS, TID, and LUN address values with an asterisk (*) represent a gap in the address assignments, or are the next available address in the run. For example, to list the available address information, enter:

```
symcfg list -dir all -address -available -sid 097
```

```
Symmetrix ID: 000190100097
```

Director				Device Name	Attr	Address		
Ident	Symbolic	Port	Sym	Physical		VBUS	TID	LUN
FA-3A	03A	0	0E2A	/dev/vx/rdmp/c5t0d0s2		0	00	000
			0E2B	/dev/vx/rdmp/c5t0d1s2		0	00	001
			04AA	/dev/vx/rdmp/c5t0d2s2	(M)	0	00	002
			04AE	/dev/vx/rdmp/c5t0d3s2	(M)	0	00	003
			04B2	/dev/vx/rdmp/c5t0d4s2	(M)	0	00	004
			04B6	/dev/vx/rdmp/c5t0d5s2	(M)	0	00	005
			04BA	/dev/vx/rdmp/c5t0d6s2	(M)	0	00	006
			04BE	/dev/vx/rdmp/c5t0d7s2	(M)	0	00	007
			04C2	/dev/vx/rdmp/c5t0d8s2	(M)	0	00	008
			04C6	/dev/vx/rdmp/c5t0d9s2	(M)	0	00	009
			-	AVAILABLE		0	00	00A *
< . . . >								

Taking RA directors offline

One or all RA director(s) of a specified Symmetrix array can be taken offline. For example, to take RA-12 of Symmetrix 097 offline, enter:

```
symcfg offline -RA 12 -sid 097
```

To bring the director back online, enter:

```
symcfg online -RA 12 -sid 097
```

Taking front-end director ports offline

A specific port of a front-end director can be taken offline as well. For example, to take port 1 of SA-12 in Symmetrix 097 offline, enter:

```
symcfg offline -SA 12 -P 1 -sid 097
```

**CAUTION**

Note: If you turn off the only connection from your host to your Symmetrix array, you will not be able to contact the Symmetrix array in order to turn it back on (online).

If you have turned off the only connection from your host, you can use another host, if one exists, to bring the director port back online. For example, to bring port 1 of SA-12 in Symmetrix 097 back online, enter:

```
symcfg online -SA 12 -P 1 -sid 097
```

Symmetrix external locks

Symmetrix external locks are used by SYMAPI (locks 0 to 15) and also for applications assigned by EMC (>15) to lock access to the entire Symmetrix array during critical operations. (Base SRDF® operations use lock 0 and the Optimizer uses lock 13.) Using the `symcfg list -lockn` parameters, you can list all locks on one or all arrays or just those locks targeted to specific operations.

A list can be returned for all, a specific number, only RDF, only RDF-A, only MSCS, or only GNS locks (ALL, #, RDF, RDFA, SRDF_MSCS, and GNS values respectively).

All possible locks

To return a list of all host-visible Symmetrix arrays (local and remote), along with details about all Symmetrix exclusive locks (known to SYMAPI) that are currently locked, enter:

```
symcfg list -lockn all
```

The following is an example output:

S Y M M E T R I X L O C K S					
SymmID	Attachment	Lock Status	Lock Number	Lock Usage	Time Held (Sec)
000000006196	Local	Unknown	N/A	N/A	N/A
000184600063	Local	Unknown	N/A	N/A	N/A
000184600282	Local	Unknown	N/A	N/A	N/A
000187900039	Remote	EXCLUSIVE	15	ConfigChg	307

In the example above, the returned list contains three local Symmetrix arrays that have no known locks, as specified by the N/A values. However, remote Symmetrix 000187900039 has an exclusive lock number 15 for a configuration change activity.

Details about a specific lock number

To focus your returned data on one or all lock number(s) on one or all Symmetrix array(s) use the `-lockn` option. For example, to return a list of all host-visible Symmetrix arrays (local and remote), along with details about any lock 0 that are currently locked, enter:

```
symcfg list -lockn 0
```

S Y M M E T R I X L O C K S					
SymmID	Attachment	Lock Status	Lock Number	Lock Usage	Time Held (Sec)
000000006196	Local	Unknown	N/A	N/A	N/A
000184600063	Local	Unknown	N/A	N/A	N/A
000184600282	Local	Unknown	N/A	N/A	N/A
000187900039	Remote	Unknown	N/A	N/A	N/A

In the example above, the returned list shows that none of the Symmetrix arrays currently have lock 0 locked, as specified by N/A.

Note that Symmetrix 000187900039 still holds lock 15 from the previous example, however, the command line query specifically requested information on exclusive lock 0, so the returned data for a Symmetrix array that held any other lock number would be returned as N/A.

Note: To return information on all known Symmetrix exclusive locks currently locked, use the keyword ALL with the `-lockn` option. When using `symcfg list -lock`, where the `-lockn` option is omitted, the lock number 0 (base SRDF operations) is assumed.

Details about a specific lock

To focus your returned data on all possible locks use the keyword `ALL`, for RDF-related locks use the keyword `RDF`, for RDF-A-related locks use the keyword `RDFA`, for MSCS related locks use the keyword `SRDF_MSCS`, and for GNS related locks use the keyword `GNS` with the `-lockn` option. In addition, you can specify a specific lock number. For example, to return information about lock 23, enter:

```
symcfg list -lockn 23 -sid 190
```

```
Symmetrix ID: 000000006190
```

```

      S Y M M E T R I X      L O C K S

SymmID      Attachment      Lock      Lock      Lock      Time
000000006190 Local      Status      Number      Usage      Held (Sec)
                                Locked      23      Unknown      20

```

In addition, the verbose option (`-v`) can be used to provide, if available from the base daemon, extended lock information about the application and host that owns the lock. For example:

```
symcfg list -lockn 23 -sid 190 -v
```

```
Symmetrix ID: 000000006190
```

```

Symmetrix ID           : 000000006190   (local)
Lock Number            : 23
Lock Usage              : Unknown
Time Held in Seconds    : 23
Lock Holder ID         : 0x1010002

Initiator               : 0
Director Number         : 1
Logical Path ID         : -1

Vendor ID              : EMC Corp
Application ID          : API_Interactive
Application Rev         : 6.3.0 (Edit Level: 762)
Host ID                : USELJTILES1
Client ID              : N/A
User Name              : ENG/lauraj
User Data               : EMC test program
Process ID             : 2644
SYMAPI Version         : V6.3.0.17 (Edit Level: 762)

```

Release a lock

SYMCLI provides the ability to release a lock on a Symmetrix array. This is not a recommended procedure and is only useful for locks which you have confirmed are stranded. For example, to release an external lock 0, which has confirmed to be stranded on Symmetrix 097, enter:

```
symcfg -sid 097 -lockn 0 release
```

**CAUTION**

Use the release lock action only if you believe that the Symmetrix lock was forgotten and there are NO other operations in progress to the specified devices in the Symmetrix array (local or remote). In addition, make sure that your application is authorized to use the lock number you are specifying. It is rarely necessary to perform this operation on any lock15 since this lock is associated with configuration change sessions and releasing it can cause problems with your processing of your configuration changes.

For information about device external locks that only target specific devices, refer to [“Releasing device external locks” on page 3-32](#).

SYMAPI semaphores

To display gatekeeper (GK), database (DB), and lock file (FILE) semaphores, enter:

```
symcfg list -semaphores
```

```

                S Y M A P I      S E M A P H O R E S

ID      Key      State      Lock Proc.
-----
5 0x45000339 Unlocked GK      0 /dev/rdsk/c5t0d0s2
5 0x45000339 Unlocked GK      0 /dev/rdsk/c5t1d21s2
3 0x4500307d Unlocked GK      0 /dev/vx/rdmp/c4t1d0s2
6 0x4500334d Unlocked GK      0 /dev/vx/rdmp/c4t1d1s2
524298 0x45003634 Unlocked GK      0 /dev/vx/rdmp/c4t1d199s2
524300 0x4500391b Unlocked GK      0 /dev/vx/rdmp/c4t1d200s2
4 0x450036af Unlocked GK      0 /dev/vx/rdmp/c5t0d0s2
8 0x45003996 Unlocked GK      0 /dev/vx/rdmp/c5t0d1s2
3 0x4500307d Unlocked GK      0 /dev/vx/rdmp/c5t0d2s2
6 0x4500334d Unlocked GK      0 /dev/vx/rdmp/c5t0d3s2
524298 0x45003634 Unlocked GK      0 /dev/vx/rdmp/c5t0d4s2
524300 0x4500391b Unlocked GK      0 /dev/vx/rdmp/c5t0d5s2
7 0x450030d5 Unlocked GK      0 /dev/vx/rdmp/c5t1d0s2
524299 0x450033a5 Unlocked GK      0 /dev/vx/rdmp/c5t1d1s2
2 0x45000ef6 Unlocked DB      0 /var/symapi/db/symapi_db.bin
65552 0x45000ee6 Locked  FILE      1 /var/symapi/config/symapinlck
1 0x45000ef1 Unlocked FILE      0 /var/symapi/config/symapislck

```

LRU cache management group

All, or a specific, least-recently-used (LRU) cache management group(s) for one or all Symmetrix array(s) can be obtained via the `-lru` option of the `symcfg list` command. For example, to view a list of all LRUs for the Symmetrix array 000000006196 enter:

```
symcfg list -lru all -sid 6196
```

The following is an example output:

```
Symmetrix ID: 000000006196 (Local)
```

LRU Num	LRU Name	Cache Slots	Percent of Total
-----	-----	-----	-----
1	GROUP_0	11971	4%
2	GROUP_1	11971	4%
3	GROUP_2	11971	4%
4	GROUP_3	11971	4%
5	GROUP_4	11971	4%
6	GROUP_5	11971	4%
7	GROUP_6	11971	4%
8	GROUP_7	11971	4%
9	GROUP_8	11971	4%
10	GROUP_9	11971	4%
11	GROUP_A	11971	4%
12	GROUP_B	11971	4%
13	GROUP_C	11971	4%
14	GROUP_D	11971	4%
15	GROUP_E	11971	4%
16	GROUP_F	11971	4%
	-----	-----	
	Total	278616	

The output returned includes a table listing the LRU number and name, the cache slots that LRU occupies, and the percentage of the total cache. In addition, the total number of cache slots occupied by the listed LRUs is provided at bottom.

Viewing network services

To view the configured network services available for use by the SYMAPI client connection, enter:

```
symcfg list -services
```

These services are also listed in the network services file (`netcnfg`) located in the `/var/symapi/config` directory.

Viewing memory board information

SYMCLI provides the ability to view memory board information per Symmetrix array. This information includes the number of boards, the slot number, and the capacity information in MBs. To view the available memory board information for all Symmetrix arrays, enter:

```
symcfg list -memory
```

Mainframe CU image

If your storage environment contains devices mapped to either EA (ESCON) or EF (FICON) front-end directors, you can utilize the `symcfg` command to return mainframe CU (Controller Unit) image information. Since devices in the mainframe environment are managed with respect to the CU image that they are a part of, SYMAPI has created a view of the CU images that are defined within the Symmetrix array. A CU image definition includes the SSID assigned to the image, the front end ports to which it is mapped, the devices included in the image, and their base and alias addresses. It also indicates whether it uses dynamic or static PAV (parallel access volumes) and whether the CU is online or not.

To view a list of CU images for all Symmetrix arrays (you can optionally limit the return data to one array by specifying the Symmetrix ID), use the following command:

```
symcfg list -cuimage
```

From the list of returned CU images, you can obtain details about a specific image using the `show` command:

```
symcfg show -cuimage 0 -ssid_num 640
```

The example above makes use of the `-ssid_num` option to return the specified CU image located on a specific sub-system ID in the case where multiple CU images with the number 0 may exist across multiple subsystems.

Environmental data

For arrays running Enginuity 5771 or higher, you can return the status of the major hardware modules including fans, power supplies, drive enclosures, and link control cards. When querying for environmental data, you must specify an `-sid`. To return an overall status for all environmental components of a given array, enter:

```
symcfg -sid 097 list -env_data
```

```
Symmetrix ID           : 000190100097
Timestamp of Status Data : 09/29/2005 12:09:54
```

System Bay

```
Bay Name                : SystemBay
Number of Fans           : 3
Number of Power Supplies : 8
Number of Standby of Power Supplies : 8
```

Summary Status of Contained Modules

```
All Fans                : Normal
All Power Supplies       : Normal
All Standby Power Supplies : Failed (1)
```

Drive Bays

```
Bay Name                : Bay-1A
Number of Standby of Power Supplies : 8
Number of Drive Enclosures : 16
```

Summary Status of Contained Modules

```
All Enclosures          : Normal
All Link Control Cards   : Normal
All Power Supplies       : Normal
All Standby Power Supplies : Normal
```

```
Bay Name                : Bay-1B
```

```

Number of Standby of Power Supplies : 8
Number of Drive Enclosures          : 16

```

```

Summary Status of Contained Modules
All Enclosures                     : Normal
All Link Control Cards             : Normal
All Power Supplies                 : Normal
All Standby Power Supplies         : Normal

```

To return a detailed status for each environmental components of a given bay (in this example, the bay name is `SystemBay`) of a specified Symmetrix, enter:

```
symcfg -sid 097 show -env_data SystemBay
```

```

Symmetrix ID           : 000190100097
Timestamp of Status Data : 09/29/2005 12:13:48

```

System Bay

```

Bay Name                : SystemBay
Number of Fans           : 3
Number of Power Supplies : 8
Number of Standby of Power Supplies : 8

```

Status of Contained Modules

```

Fans
FAN-01                  : Normal
FAN-02                  : Normal
FAN-03                  : Normal

```

```

Power Supplies
PS-01                   : Normal
PS-02                   : Normal
PS-03                   : Normal
PS-04                   : Normal
PS-05                   : Normal
PS-06                   : Normal
PS-07                   : Normal
PS-08                   : Normal

```

```

Standby Power Supplies
SPS-1A                  : Normal
SPS-2A                  : Normal
SPS-3A                  : Normal
SPS-4A                  : Normal
SPS-4B                  : Normal
SPS-3B                  : Normal
SPS-2B                  : Normal
SPS-1B                  : Failed

```

Notice in the output that the last entry of Standby Power Supplies, SPS-1B, has a status of failed.

The `symcfg` command provides options that allows the returned data to be limited to a specific service state (`-service_state` option) — either degraded, failed, or normal—, or all service states except one by preceding the service state value with a `not`, such as `-service_state notfailed`.

For example, to return the environmental data for Symmetrix 097 with a service state of failed, enter:

```
symcfg -sid 097 list -env_data -service_state failed
```

```
Symmetrix ID           : 000190100097
Timestamp of Status Data : 09/29/2005 12:15:34
```

System Bay

```
Bay Name                : SystemBay
Number of Fans           : 3
Number of Power Supplies : 8
Number of Standby of Power Supplies : 8
```

Status of Contained Modules

```
Fans
Power Supplies
Standby Power Supplies
SPS-1B                  : Failed
```

Drive Bays

```
No components are in the specified state
```

Notice that the returned data only contains information about the bay containing the failure, and the components in the failed state.

Symmetrix devices

From the perspective of software running on a host system, a Symmetrix array appears to be a number of *physical devices* connected to one or more I/O controllers. A host application addresses each of these devices using a physical device name. Each physical device defined in the configuration database has a specific set of attributes (such as vendor ID, product ID, revision level, and serial ID).

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 is mapped is called a *hypervolume* or a *hyper*. A Symmetrix device may map to multiple hypers (containing identical copies of data) depending on its mirror configuration.

The Symmetrix database file maintains device-level configuration and status information for each device on every Symmetrix array that is accessible from the host. Using SYMCLI, you can obtain a list of all the available devices. You can then use the returned device data from this list to obtain its configuration and status information. This information can lead you to back-end information for the device's disk director(s) and corresponding hypervolumes, and their mappings to disk drives.

Device emulation

All host I/O transactions with an array of Symmetrix disk devices are managed by the Enginuity operating environment, which runs in the Symmetrix I/O subsystem (channel directors and disk directors). Because each of the physical disks are indirectly seen as part of the I/O protocol, Symmetrix devices are presented to the host with the following configuration or emulation attributes:

- ◆ Each device has **N** cylinders. The number is configurable ($\text{blocks} \div 960$).
- ◆ Each cylinder has **15** tracks (heads).
- ◆ Each device track in a fixed block architecture (FBA) has **64** blocks of **512** bytes. (For non-FBA operating systems, the blocks are recognized without regard to the number of bytes.)

Note: With Enginuity 5771, track sizes for FBA devices has been increased to 64K.

Device type overview

The SYMAPI provides the ability to define and configure devices for numerous specialized roles defined as *device types*. Each device type has specialized characteristics that enables a device to participate in various SYMAPI operations. The following tables describes the device types of the Symmetrix storage environment.

Table 3-1 Device types

Device type	Description
Standard Devices	A Symmetrix device configured for normal Symmetrix operation under a desired protection method (such as RAID 1, RAID-S, and SRDF).
Gatekeeper Devices	SCSI commands executed by SYMAPI are transferred to the Symmetrix array via a Symmetrix device that is designated as a <i>gatekeeper</i> device. The gatekeeper allows you to retrieve configuration and status information from the Symmetrix array without interfering with normal device I/O operations. By default, one of the available Symmetrix devices is designated as a gatekeeper. Alternatively, you can define specific devices to be used as gatekeepers. For more information, refer to Chapter 10, "Gatekeeper Devices."
Meta Devices	Allow individual devices to be concatenated to create larger devices. A meta device consists of a meta head and its member devices. The <i>meta head</i> is the first device in the meta device sequence and is responsible for receiving all incoming commands. It also identifies the entire meta device. When an incoming command for the meta head is processed, the Symmetrix determines which meta device member should execute the command. Meta head devices can be added to a device group while a <i>meta member</i> cannot be added to a device group. This holds true for both standard and BCV devices.
BCV Devices	Specialized devices used to create a local copy of data contained in a standard Symmetrix device, which can be used for backup, restore, decision support, and application testing. Symmetrix TimeFinder is a business continuance solution that allows you to use these special Symmetrix devices called BCV devices (Business Continuance Volume). Each BCV device has its own host address, and is configured as a stand-alone Symmetrix device. For more information, refer to the <i>EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide</i> .
SRDF Devices	Devices configured as RDF1 or RDF2 to support SRDF operations. The Symmetrix Remote Data Facility (SRDF) is a business continuance solution that maintains a device-level mirror of Symmetrix data on remotely attached Symmetrix arrays. These arrays also may be located in physically separate sites. SRDF provides a recovery solution for component or site failures using remotely mirrored devices. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster. For more information, refer to the <i>EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide</i> .
Dynamic RDF Devices	Since Engenuity 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, 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 via the Configuration Manager and the devices must be designated as Dynamic RDF-capable devices. For information about Dynamic RDF devices, refer to the <i>EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide</i> .
Virtual Devices	A host-accessible device containing track-level location information (pointers), which indicates where the copy session data is located in the physical storage. Device copies use <i>virtual devices</i> to support TimeFinder/Snap operations. 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. For more information, refer to the <i>EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide</i> .

Table 3-1 Device types (continued)

Device type	Description
Save Devices	Special devices (not mapped to the host) that provide physical storage space for pre-update images or changed tracks during a virtual copy session of TimeFinder/Snap operations. Save devices are a predefined pool of storage devices and must be configured for this purpose. The save device pool acts as a group for storing data in striped form. Save devices are assigned a Symmetrix device number and can be unprotected, mirrored, or parity RAID. For more information, refer to the <i>EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide</i> .
Device Masking (VCM) Devices	Symmetrix devices that have been masked for visibility to certain hosts. The device masking database (VCMDB) holds device masking records and typically resides on a 24 or 48 cylinder disk device. For more information, refer to the <i>EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide</i> .
DRV Devices	A non-user-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. For more information, refer to the <i>EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide</i> .

List of devices

Primarily, there are two commands to obtain a list of devices:

- ◆ `sympd` — Lists Symmetrix devices that are host-visible.
- ◆ `symdev` — Displays information about all Symmetrix devices, host-visible or those not necessarily visible to your host.

Use `sympd` to list all physical device (`pd`) names (in the left column) with the corresponding Symmetrix device (`dev`) names (in the next column) for all devices (visible to your host) on all Symmetrix arrays.

For example, for a list of physical device names, enter:

```
sympd list
```

The corresponding output shows the physical device (`Physical`) names, the corresponding Symmetrix device names (`Sym`), director information, and device-specific information for all devices (visible to your host) on all Symmetrix arrays. The following is an example output for one Symmetrix array:

```
Symmetrix ID: 000000006196
```

Device Name	Directors			Device			
Physical	Sym	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)
/dev/rdsk/clt0d0s2	0000	02A:1	01C:C6	Unprotected	N/Grp'd	VCM WD	47
/dev/rdsk/clt0d1s2	0040	02A:1	16A:D4	Unprotected	N/Grp'd	RW	188
/dev/rdsk/clt0d2s2	0041	02A:1	01D:D4	Unprotected	N/Grp'd	RW	188
/dev/rdsk/clt0d3s2	0042	02A:1	16B:C4	Unprotected	N/Grp'd	RW	188
/dev/rdsk/clt0d4s2	0043	02A:1	01A:C4	Unprotected	N/Grp'd	RW	188

The `symdev` command provides similar output as the `sympd` command, but includes all Symmetrix devices and lists them by Symmetrix device names for all devices on all or a given Symmetrix array(s). The corresponding output shows the Symmetrix device names (*Sym*), physical device (*Physical*) names, director information, and device-specific information for all devices on the Symmetrix array.

Note: The physical device names are not known for those devices that are not visible to the host making the request. A value of `??? :` for the SA means there is no mapping to a front-end director port. A value of `*** :` means there are multiple mappings.

For example, for a list of Symmetrix device names on Symmetrix 097 enter (the example output that follows is truncated):

```
symdev list -sid 097
```

```
Symmetrix ID: 000190100097
```

Device Name		Directors		Device			
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)	
04AA	/dev/vx/rdmp/c5t0d2s2	03A:0 01A:C2	2-Way Mir	Grp'd	(M) RW	17261	
04AE	/dev/vx/rdmp/c5t0d3s2	03A:0 01D:C3	2-Way Mir	Grp'd	(M) RW	17261	
04B2	/dev/vx/rdmp/c5t0d4s2	03A:0 16C:D2	2-Way Mir	Grp'd	(M) RW	17261	
04B6	/dev/vx/rdmp/c5t0d5s2	03A:0 16C:C3	2-Way Mir	Grp'd	(M) RW	17261	
04BA	/dev/vx/rdmp/c5t0d6s2	03A:0 01A:C4	RAID-5	Grp'd	(M) RW	17261	
04BE	/dev/vx/rdmp/c5t0d7s2	03A:0 01A:D5	RAID-5	Grp'd	(M) RW	17261	
04C2	/dev/vx/rdmp/c5t0d8s2	03A:0 06C:C7	RAID-5	Grp'd	(M) RW	17261	
04C6	/dev/vx/rdmp/c5t0d9s2	03A:0 01A:C4	RAID-5	Grp'd	(M) RW	17261	
0E2A	/dev/vx/rdmp/c5t0d0s2	03A:0 15D:CD	Unprotected	N/Grp'd	RW	6	
0E2B	/dev/vx/rdmp/c5t0d1s2	03A:0 15D:CD	Unprotected	N/Grp'd	RW	6	

```
< . . . >
```

**Example output
verbose option**

The verbose option (-v) with either `symdev list` or `sympd list` will return details about each device (for a single device's details use the `show` argument). The following is an example of the detailed output returned for a host-visible device using the `symdev show -v` command entry:

```
symdev show 04B2 -sid 097
```

```
Symmetrix ID: 000190100097
```

```
Device Physical Name      : /dev/vx/rdmp/c5t0d4s2
```

```
Device Symmetrix Name    : 04B2
```

```
Device Serial ID        : 97004B2000
```

```
Symmetrix ID            : 000190100097
```

```
Device Group Name       : bp
```

```
Device Logical Name     : DEV003
```

```
Attached BCV Device     : N/A
```

```
Attached VDEV TGT Device : N/A
```

```
Vendor ID               : EMC
```

```
Product ID              : SYMMETRIX
```

```
Product Revision        : 5771
```

```
Device WWN              : 60060480000190100097533030344232
```

```
Device Emulation Type   : FBA
```

```
Device Defined Label Type: N/A
```

```
Device Defined Label    : N/A
```

```
Device Sub System Id    : 0x0005
```

```
Device Block Size       : 512
```

```
Device Capacity
```

```
{
  Cylinders      :      36824
  Tracks         :      552360
  512-byte Blocks :    35351040
  MegaBytes      :      17261
  KiloBytes      :    17675520
}
```

```
Device Configuration    : 2-Way Mir      (Meta Head,
                                         Non-Exclusive Access)
```

```
Device is WORM Enabled  : No
```

```
Device is WORM Protected : No
```

```
SCSI-3 Persistent Reserve: Disabled
```

```
Dynamic Spare Invoked   : No
```

```
Dynamic RDF Capability   : RDF1_OR_RDF2_Capable
```

```
STAR Mode               : No
```

```
STAR Recovery Capability : None
```

```
STAR Recovery State     : NA
```

```
Device Service State    : Normal
```

```
Device Status           : Ready          (RW)
```

```
Device SA Status        : Ready          (RW)
```

```
Front Director Paths (3):
```

```
{
```

```
-----
```

PdevName	POWERPATH Type	DIRECTOR Type Num	PORT Sts	VBUS	TID	SYMM	LUN Host
/dev/vx/rdmp/c5t0d4s2	N/A	FA 03A:0	RW	000	00	004	004
/dev/rdisk/c5t0d4s2	N/A	FA 03A:0	RW	000	00	004	004
Not Visible	N/A	FA 07A:0	RW	000	00	044	N/A

Meta Configuration : Striped
 Meta Stripe Size : 960k (2 Cylinders)
 Meta Device Members (4) :
 {

BCV DATA					RDF DATA		
Sym Dev	Cap (MB)	Std Inv Tracks	BCV Inv Tracks	Pair State	R1 Inv Tracks	R2 Inv Tracks	Pair State
--> 04B2	4315	-	-	N/A	-	-	N/A
04B3	4315	-	-	N/A	-	-	N/A
04B4	4315	-	-	N/A	-	-	N/A
04B5	4315	-	-	N/A	-	-	N/A
17261		-	-		-	-	

 }
 Mirror Set Type : [Data,Data,N/A,N/A]
 Mirror Set DA Status : [RW,RW,N/A,N/A]
 Mirror Set Inv. Tracks : [0,0,0,0]
 Back End Disk Director Information
 {
 Hyper Type : Data
 Hyper Status : Ready (RW)
 Disk [Director, Interface, TID] : [16C, D, 2]
 Disk Director Volume Number : 661 (0x294)
 Hyper Number : 68
 Disk Capacity : 285902m
 Disk Group Number : 2

 Hyper Type : Data
 Hyper Status : Ready (RW)
 Disk [Director, Interface, TID] : [06A, D, 2]
 Disk Director Volume Number : 665 (0x298)
 Hyper Number : 65
 Disk Capacity : 285902m
 Disk Group Number : 2
 }

Device service states

There are three possible device service states:

- ◆ Normal
- ◆ Failed
- ◆ Degraded

Note: Degraded indicates that one or more mirror positions to the protected device is Not Ready.

Device status

There are three possible states to the device status:

- ◆ Ready
- ◆ Not Ready
- ◆ Write Disabled

You can change the status of a device or a group of devices (or by specifying a file containing a list of devices using `-file <FileName>`) using the following `symdev` command options:

```
symdev -sid SymmID [-noprompt] [-celerra] [-star]
  rw_enable      SymDevName [-SA <#|ALL> [-P #]]
  write_disable  SymDevName [-SA <#|ALL> [-P #]]
  ready          SymDevName
  not_ready      SymDevName
  relabel        SymDevName
  hold           SymDevName
  unhold         SymDevName
```

```
symdev -sid SymmID -file <FileName> [-noprompt] [-celerra] [-star]
  rw_enable      [-SA <#|ALL> [-P #]]
  write_disable  [-SA <#|ALL> [-P #]]
  ready
  not_ready
  relabel
  hold
  unhold
```

Device emulation type

To return a table that provides an inventory of configured devices by emulation type, use the following command, followed by the output:

```
symdev list -inventory
```

```
Symmetrix ID: 000187900771
```

Device Config	FBA	CKD	AS400	CELERRA
-----	----	----	----	-----
Unprotected	1140	1	N/A	N/A
2-Way Mir	397	N/A	N/A	N/A
3-Way Mir	1	N/A	N/A	N/A
RAID-S	3	N/A	N/A	N/A
RAID-5	1115	1	N/A	N/A
RDF2	26	N/A	N/A	N/A
RDF2+R-S	3	N/A	N/A	N/A
BCV	595	N/A	N/A	N/A
DRV	2	N/A	N/A	N/A
VDEV	8	N/A	N/A	N/A
COVD	26	N/A	N/A	N/A

This list can be limited to the devices in a specific Symmetrix array by specifying a Symmetrix ID (-sid).

Filtering device data with symdpd

When calling `symdpd list`, a number of filters can be utilized to focus your list output of host-visible devices. These options are all relative to the host from which the request is generated.

When calling this command, a number of filter options can be specified to reduce the output of the command to the specific types of devices you're investigating. Limit the device output to a specific Symmetrix array by specifying the `-sid` option. List host-visible Symmetrix devices that match a specific DA (`-da`), interface (`-interface`), disk (`-disk`) or hyper-volume (`-hyper`) values. The interface, disk, and hyper values default to ALL if they are not specified.

You can also list the host-visible Symmetrix devices that match a specific front-end director number (`-sa`) or front-end director port number (`-p #`). The output can be further limited to devices that are mapped to SCSI front-end directors (`-scsi`), or Fibre front-end directors (`-fibre`).

While the `symdpd` command only provides a limited amount of filter options, additional filter options on physical devices can be found by utilizing the `symdev` command with the `list pd` argument and associated options. Refer to ["Filtering Symmetrix device data" on page 3-30](#) for more information.

Device output options

Specifying the `-cyl` option modifies the output list to include device capacity in cylinders rather than the default of megabytes (MB).

Adding specific device types

Additionally, options can be added to include specific device types and information. The `-powerpath` option can be used to include the host-visible EMC® PowerPath® devices, their Symmetrix configurations, and their alternate paths. Adding the `-vcm` option will include Volume Logix (VCM) devices in the output list.

Filtering Symmetrix device data

The `symdev list` and `symdev list pd` returns a list of all Symmetrix devices that are configured in one or more Symmetrix arrays connected to the host from which the request was generated. The `pd` qualifier has been added to the list action to return only physical devices. The `pd` qualifier must appear after the list action on the command line and can be used with all other list options as well.

When calling this command, a number of filter options can be utilized to focus your list output of all devices. These options are all relative to the host from which the request is generated. The following options are available:

```
[ -sid SymmID] [ -offline] [ -v] [ -resv] [ -wwn] [ -all]
[ -SA <#|ALL>] [ -P #] [ -scsi] [ -fibre]
[ -RANGE <SymDevStart>:<SymDevEnd>] [ -R1] [ -R2] [ -cyl]
[ -CAP #] [ -N #] [ -vcm] [ -held] [ -gige] [ -ficon] [ -escon]
[ -noport] [ -firstport] [ -multiport] [ -bcv] [ -nobcv] [ -drv]
[ -meta] [ -hotspare] [ -dynamic] [ -worm] [ -vdev]
[ -savedev] [ -raids] [ -rdfa] [ -disk_group nn] [ -raid5]
[ -emulation fba|ckd|as400|celerra] [ -star_mode]
[ -star_sync_target] [ -star_async_target]
[ -half_pair] [ -dup_pair] [ -bcv_emulation]
[ -reserved | -noreserved]
[ -service_state [no]degraded | [no]failed | [no]normal]
```

In all `symdev list` and `symdev list pd` command entries, you can limit the device output to a specific Symmetrix array by specifying the `-sid` option. The following provides additional details about some of the options.

Using the `-all` option

The `symdev` command has an `-all` option, which returns all configured devices, including private devices such as DRV, Vault, COVD, SAVEDEV, and SFS devices. Prior to Solutions Enabler V6.0, all devices were returned (including private devices) by default when no options were specified. However, in Solutions Enabler V6.0 and higher, the `-all` option is required to return all configured devices. Therefore, scripts written prior to Solutions Enabler V6.0 that utilize `symdev list` and `symdev list pd`, must be upgraded to include the `-all` option where all configured devices are required.

Filter by director

You can list the host-visible Symmetrix devices that match a specific front-end director number (`-SA`) or front-end director port number (`-P #`). The output can be further limited to devices that are mapped to SCSI front-end directors (`-scsi`), or Fibre front-end directors (`-fibre`).

Filter by director port mapping

In addition, you can limit the returned list of Symmetrix devices to one of the following: devices that are mapped to more than one front-end director port (`-multiport`), devices that are not mapped to any front-end director port (`-noport`), or first port information for devices that have more than one port mapping schemas (`-firstport`). If none of these options are specified, then devices with all director-port relationships are returned.

Filter by device type

Additional options are available to filter your return data by device type. For more information on `symdev` device type options, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

RAID device information

You can list RAID devices by RAID group using the `-raids` (or `-raid`, which performs the same operation) option

When viewing device output for RAID-5 devices, it is important to note that the total device capacity (`Device Capacity`) displayed is actually the total logical capacity. Whereas, the hyper capacity (`Hyper Num Cap`) displayed for each member is the physical capacity, which include all device space — available and unavailable. The truncated example output below highlights the differences:

```
Device Physical Name      : Not Visible

Device Symmetrix Name    : 118B
Device Serial ID         : N/A
Symmetrix ID             : 000000006196
WWN                      : 6006048000000000619653594D483842

. . .

Device Capacity
{
  Cylinders      :      100
  Tracks         :      1500
  512-byte Blocks :      96000
  MegaBytes      :          47
  KiloBytes      :      48000
}

. . .

RAID-5 Device Information
{
  Number of Tracks in a Stripe      : 4
  Overall Ready State of RAID-5 Device : ReadyNoSpare
  Overall WriteProtect State of RAID-5 Device : EnabledNoSpare
  Member Number of the Failing Device : None
  Mirror Number of the Failing Device : None
  Mirror Number of the Data Device    : 0
  Member Number that Invoked the Spare : None
  Mirror Number of the Spare Member   : None
  Disk Director (DA) that Owns the Spare : None
  Copy Direction                     : N/A
  RAID-5 Hyper Devices (3+1):
  {
    Device : 118B
    {
      -----
      Disk    DA      Hyper      Member      Spare      Disk
      DA :IT  Vol#    Num Cap(MB)  Num Status  Status  Grp#  Cap(MB)
      -----
      16A:D2   443    72      16      4 RW      N/A      2   139814
      01B:C3   146    72      16      1 RW      N/A      2   139814
      01C:C6   296    72      16      3 RW      N/A      0   139814
      16D:D7   590    71      16      2 RW      N/A      2   139814
    }
  }
}
```

Filter by DA, interface, disk, or hypervolume

List host-visible Symmetrix devices that match a specific DA (`-da`), interface (`-interface`), disk (`-disk`), disk group number (`-disk_group`), or hyper-volume (`-hyper`) values. The interface, disk, and hyper values default to ALL if they are not specified.

Device output options

Specifying the `-cyl` option modifies the output list to include device capacity in cylinders rather than the default of megabytes (MB). The `-pd` option will only display the host visible devices (PDevs).

Detailed device configuration data

You can use two commands to gather information about a specific device:

- ◆ `sympd` — displays information about a Symmetrix device that is visible to your host.
- ◆ `symdev` — displays information about Symmetrix devices not necessarily visible to your host.

They both provide nearly the same information. For example on UNIX, to get information about device `/dev/rdisk/c3t1d1s2`, which is Symmetrix device 0046, enter either:

```
sympd show /dev/rdisk/c3t1d1s2
symdev show 0046
```

On Windows, to get information about device `\\.\physicaldrive2`, enter:

```
sympd show physicaldrive2
```

The output is the same as the verbose option of `symdev list`. To view an example output, refer to [“Example output verbose option” on page 3-26](#).

Note: The `symdev show` and `sympd show` commands will always display a zero for TID with FA's configured on the same Symmetrix array.

Releasing device external locks

The EMC Solutions Enabler uses *device external locks* in the Symmetrix to lock pairs during replication operations (such as Open Replicator, TimeFinder, and SRDF operations).

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

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

If you discover a lock from your host that has been held for well over 2 hours and you have confirmed no one is using the device resources, you can choose to release the lock. To release the device lock on a range of Symmetrix devices in Symmetrix 870, enter:

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



CAUTION

Use the release lock action only if you believe that the Symmetrix device lock was forgotten and there are NO other operations in progress to the specified Symmetrix devices (local or remote). Locks are typically of short duration (one second to an hour or so). However, it is critical to be able to recognize when a device lock being held by certain applications (such as an RDF action) are allocated as long duration locks.

CLARiiON devices

Similar to Symmetrix devices, the `symdev -clariion list` command can be used to return data about CLARiiON devices. The returned list of CLARiiON devices can be limited to a specific CLARiiON by specifying a CLARiiON ID (`-cid`), by capacity (`-cap`), by a number of devices (`-N`), or a range of devices (`-RANGE <start:end>`).

Example: CLARiiON devices

The following example shows the output for a list of CLARiiON devices in the device number range of 230 to 250:

```
symdev -clariion list -range 230:250
```

```
Clariion ID: APM00034501433
```

Device		Device		
Num	Physical Name	Config	Cap (MB)	WWN
0230	Not Visible	RAID-5	1024	6006016095E30D005B6301EFAD10D911
0231	Not Visible	RAID-5	1024	6006016095E30D005C6301EFAD10D911
0232	Not Visible	RAID-5	1024	6006016095E30D005D6301EFAD10D911
0233	Not Visible	RAID-5	1024	6006016095E30D005E6301EFAD10D911
0234	Not Visible	RAID-5	1024	6006016095E30D00F4691DF5AD10D911
0235	Not Visible	RAID-5	1024	6006016095E30D00F5691DF5AD10D911
0236	Not Visible	RAID-5	1024	6006016095E30D00F6691DF5AD10D911
0237	Not Visible	RAID-5	1024	6006016095E30D00F7691DF5AD10D911
0238	Not Visible	RAID-5	1024	6006016095E30D00F8691DF5AD10D911
0239	Not Visible	RAID-5	1024	6006016095E30D00F9691DF5AD10D911
0240	Not Visible	RAID-5	1024	6006016095E30D00FA691DF5AD10D911
0241	Not Visible	RAID-5	1024	6006016095E30D00FB691DF5AD10D911
0242	Not Visible	RAID-5	1024	6006016095E30D00FC691DF5AD10D911
0243	Not Visible	RAID-5	1024	6006016095E30D00FD691DF5AD10D911
0244	Not Visible	RAID-5	1024	6006016095E30D00FE691DF5AD10D911
0245	Not Visible	RAID-5	1024	6006016095E30D00B0BE47FBAD10D911
0246	Not Visible	RAID-5	1024	6006016095E30D00B1BE47FBAD10D911
0247	Not Visible	RAID-5	1024	6006016095E30D00B2BE47FBAD10D911
0248	Not Visible	RAID-5	1024	6006016095E30D00B3BE47FBAD10D911
0249	Not Visible	RAID-5	1024	6006016095E30D00B4BE47FBAD10D911
0250	Not Visible	RAID-5	1024	6006016095E30D006D61F3BAF10D911

Disk-level data

The configuration database file maintains low-level configuration and status information for each disk on every Symmetrix array that is accessible from the host. Using SYMCLI, you can get a list of all the available disks. You can then choose a disk from this list and get its configuration and status information.

List and show disks

There are two `symdisk` command arguments you can use to gather disk information:

- ◆ `list` — Provides a list of available disks. You can use various options to expand and restrict your list output. Use the `-v`, `-hypers`, and `-hospares` options to obtain additional information about each disk; specifying a Disk Director (DA), Disk Interface (INT), or Disk Target ID (TID) will narrow down the return data.
- ◆ `show` — Displays detailed information about the disk(s) that match the given Disk Adapter (DA), Disk Director Interface (INT), and Disk Target ID (TID).

The following example displays list information about the disks and hypers on the specified Disk Adapter (02A):

```
symdisk list -sid 039 -da 02A
```

```
Symmetrix ID      : 000187900039
Disks Selected    : 7
```

Ident	Symb	Int	TID	Vendor	Type	Hypers	Total	Capacity (MB)	
								Free	Actual
DF-2A	02A	C	1	SEAGATE	CH146LF	9	139814	123266	140014
DF-2A	02A	C	3	SEAGATE	CH146LF	9	139814	122329	140014
DF-2A	02A	C	5	SEAGATE	CH146LF	9	139814	121171	140014
DF-2A	02A	C	7	SEAGATE	CH146LF	8	139814	125334	140014
DF-2A	02A	C	9	SEAGATE	CH146LF	8	139814	127397	140014
DF-2A	02A	C	B	SEAGATE	CH146LF	8	139814	125334	140014
DF-2A	02A	C	D	SEAGATE	CH146LF	8	139814	127397	140014

The following example shows details about the specified disk (02A:C9):

```
symdisk show 02A:C9 -sid 039
```

```
Symmetrix ID      : 000187900039
Director          : DF-2A
Interface         : C
Target ID         : 9
Disk Group Number : 0

Vendor ID         : SEAGATE
Product ID        : SX3146807FC
Product Revision  : CH146LF
Serial ID         : 3HY8AQWX

Disk Blocks       : 286339876
Block Size        : 512
Actual Disk Blocks : 286749475
Total Disk Capacity (MB) : 139814
Free Disk Capacity (MB) : 127397
Actual Disk Capacity (MB) : 140014
Hot Spare         : False

Failed Disk       : False
```

Hypers (8) :

#	Vol	Emulation	Dev	Type	Mirror	Status	Cap (MB)
4	39	FBA	002D	Data	1	Ready	2063
5	40	FBA	006B	Data	1	Ready	2063
1	36	FBA	00B0	RAID-S Data	1	Ready	2063
6	41	FBA	00D8	Data	1	Ready	2063
2	37	FBA	0115	Data	2	Ready	2063
7	42	FBA	014E	Data	1	Ready	3
3	38	FBA	015F	Data	1	Ready	2063
8	43	FBA	0177	Virtual	2	Ready	2

When virtual devices (VDEVs) are part of your disk configuration, the capacity shown for these virtual hypers reflects only space to store track tables (pointers). Typically, a low number, such as 3 MB, is needed for these devices.

Disk gaps

Using the `-gaps` option with the `syndisk list` command will list any gaps found on the disk and the `-spare_info` will tell you what disk the spare disk is substituting for, if it has been invoked.

The `syndisk show` command has a new `-gaps_only` options that will list any gaps found on the disk and not list all the hyper information.

Disk groups

When returning a list of disks, the `-by_diskgroup` option organizes the returned list of disks by disk group number. The `-disk_group` option allows you to limit the resulting list to disks belonging to a specified disk group.

Hotspares

The `syndisk -hotspare` command will exclusively return information about hotspare devices. If a hot spare is invoked against a failed disk, the `syndisk -v -spare_info` option can be used to return information about the failed disk that has been replaced. The following example shows the truncated output, where the hotspare has been invoked against disk 16B:C0:

```
syndisk list -v -spare_info

<...>
  Hot Spare                      : False

  Director                      : DF-16B
  Interface                     : C
  Target ID                     : A
  Disk Group Number             : 2

  Vendor ID                    : SEAGATE
  Product ID                   : SX3146807FC
  Product Revision              : CH146LF
  Serial ID                    : 3HY9CQVK

  Disk Blocks                   : 0
  Block Size                    : 512
  Actual Disk Blocks            : 286749475
  Total Disk Capacity (MB)      : 0
  Free Disk Capacity (MB)       : 0
  Actual Disk Capacity (MB)     : 140014
  Hypers                        : 0

  Hot Spare                     : True

  Failed Director               : DF-15A
  Failed Interface              : C
  Failed Target ID              : 0
```

The `-gaps` options on the `list` command will list gaps found between hypers on the disk, as the hypers are listed. To see short list of only the gap information, either do not specify the `-hypers` option on the `syndisk list` command, or specify the `-gaps_only` option on the `syndisk show` command.

Note: The gap sizes provided by this report are only an approximation. The report can be used as a general guide to the location and size of free space gaps, but it may not be accurate down to the last cylinder.

This chapter describes how to set up host-based Access Control and how to perform Symmetrix Access Control actions using SYMCLI.

- ◆ Symmetrix Access Control operations overview4-2
- ◆ How to create or modify Access Control data4-5
- ◆ How to create and manage access groups4-7
- ◆ How to create and manage limited access to access pools4-10
- ◆ How to create and manage access control entries4-13
- ◆ How to obtain access control information4-18
- ◆ How to release pending locked sessions.....4-19
- ◆ Grant/deny access strategies.....4-20
- ◆ How to back up and restore the access control database.....4-23

Symmetrix Access Control operations overview

Anyone with access to Symmetrix-based management software can execute any function on any Symmetrix device. Many product applications such as EMC ControlCenter, TimeFinder, SRDF, Optimizer, Resource View, Database Tuner, and various ISV products can issue management commands to any device in a Symmetrix complex. Open systems hosts can manipulate mainframe devices, Windows hosts can manipulate UNIX data, and vice versa.

Shared systems, such as these, may be vulnerable to one host, accidentally or intentionally, tampering with another's devices. To prevent this, the `symacl` command can be used by an administrator of the Symmetrix storage site to set up and restrict host access to defined sets of devices (access pools) across the various Symmetrix arrays.

Note: For information on user-based authorization, refer to [Chapter 5, "User Authorization."](#)

Access Control command

This SYMCLI component supports Symmetrix Access Control requirements. The Access Control command allows you to set up and maintain an access controlled environment over the Symmetrix resource (access pools). The command (`symacl`) sets up access control mechanisms and changes access control entries in the access control database.

For more information about the syntax of the `symacl` command, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Access Control database

A Symmetrix-based access control database (shown in [Figure 4-1](#)) contains all the mechanisms or information to govern access to Symmetrix access pools.

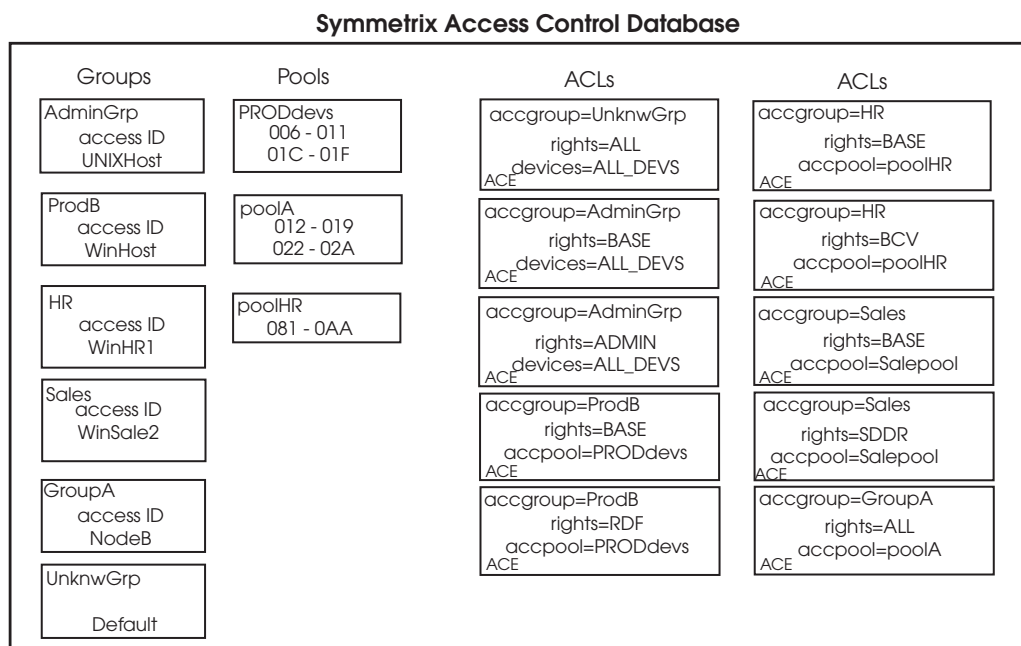


Figure 4-1 ACLs and ACEs in the Symmetrix Access Control database

Information about the following access control mechanisms comprise the access control database:

- ◆ **Access control groups** — Unique *access IDs* and *names* are assigned (together) to hosts and then sorted into access control *groups* according to similar needs (determined by an Administrator). Access groups are allowed to act on access pools based on *permissions* (*access types*) granted by the Administrator. The unique host ID for open systems can be viewed by running `symacl -unique`.
- ◆ **Access pools** — Permissions (or access types), such as BCV, SRDF, ADMIN, are assigned to allow a host to perform certain Solutions Enabler functionality on a specified set of devices. These sets of devices are referred to as access pools or `accpool`.
- ◆ **Access Control Entries (ACEs)** — Once the group and access pool mechanisms are established, the *access control entries* (ACEs) are created, which grant permissions to these pools. The ACEs of the various access groups (along with groups and pools) are managed and stored in the access control database.
- ◆ **Access Control Lists (ACLs)** — A group of ACEs that are associated with the same group. For example, in [Figure 4-1](#), the ACEs that are associated to group HR comprise an ACL in the database.

[Figure 4-2 on page 4-4](#) illustrates a Symmetrix site that is managing the permissions of various types of host connections to the Symmetrix devices. Each Symmetrix has its own access control database, which are separately managed by `symacl`.

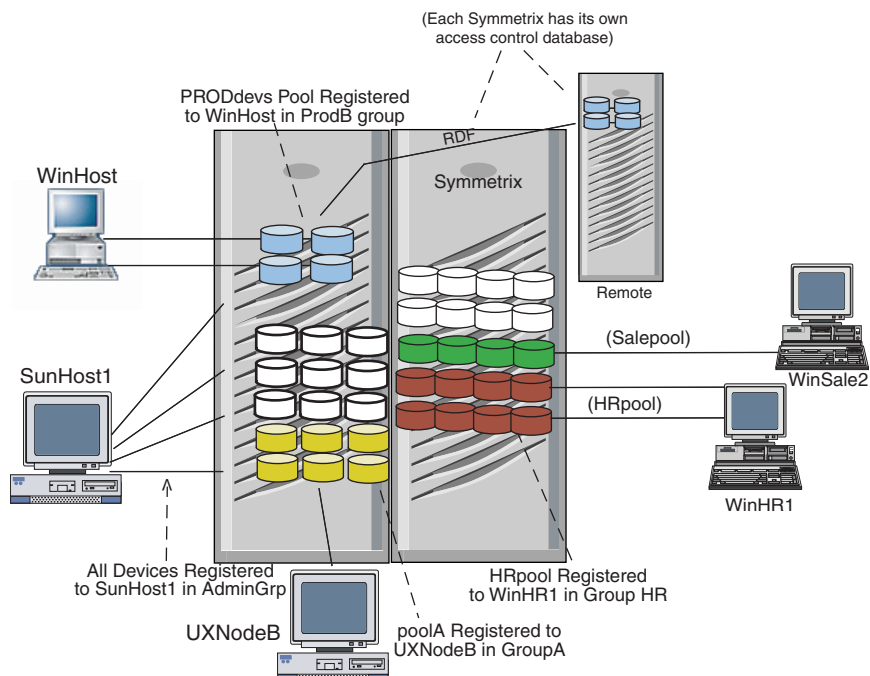


Figure 4-2 A Symmetrix site using Access Control

How to create or modify Access Control data

To create or change the Access Control information in the database, you need to enter the changes in a command file and eventually commit the file to the access control database.

The command file format contains various command entries terminated with a semicolon (;). The command syntax is parsed case insensitive, but any variable parameter entered must be case sensitive.

The following are various types of permissions creation and changes possible in the command file:

- ◆ Create new access groups
- ◆ Add and remove access IDs to access groups
- ◆ Move an access ID from one group to another
- ◆ Create new access pools
- ◆ Add and remove devices to access pools
- ◆ Delete access pools and access groups
- ◆ Add ACEs to grant access
- ◆ Remove ACEs to deny access

Note: After Access Control changes are made, you must run a discover operation (symcfg discover).

Command file execution

To safely apply any of these Access Control actions (command file entries) to the access control database, you should perform the following progressive *symacl* operations on the command file:

- ◆ Preview
- ◆ Prepare
- ◆ Commit

After you first create the command file, the *preview* operation verifies the syntax and correctness of the contents of the entries in the command file.

The *prepare* operation (also before you commit) performs the preview checks but also verifies the appropriateness of the requested access control modifications against the current state of the access control database in the Symmetrix array.

The *commit* operation performs both the preview and prepare checks and then commits the contents of the command file to the Symmetrix Access Control database.

Note: It is not mandatory to execute a preview or prepare action prior to a commit. However, these actions can ensure that the commit action will not be rejected or can be used to debug the command file entries.

Minimum Access Control configuration

Typically, following initial delivery and set up of a Symmetrix array on a site, an EMC Engineer will need to set some minimal access control parameters to enable you to begin administrative actions, such as the following example:

Table 4-1 Initial access groups set

Access groups	ID names
AdminGrp	SunHost1 ACCPIN
UnknwGrp	unknown

The name `SunHost1` shown in the [Table 4-1](#) (associated with `AdminGrp`) is the ID name that designates the machine from which you administer access controls. In order to perform prepare, commit, and release actions, an access ID (PIN) with the name `ACCPIN` was created in the delivered setup. This ID name is also used for the value setting of `SYMCLI_ACCESS_PIN`. For more information, see [“How to create and manage access groups”](#).

As shown in the following table for a delivered setup, access group `AdminGrp` has access to all devices (`ALL_DEVS`) and this group was granted both `ADMIN` and `ALL` permissions. Also in the initial setup, a group called `UnknwGrp` is established with permissions to all devices in the Symmetrix. At the outset of system usage, this gives all unknown hosts `BASE` permissions to all devices (`ALL_DEVS`) until such time it becomes clear as to what restrictions should be established.

Initial ACL setup

The initial `UnknwGrp` setup also grants `ALL` permissions to any devices not already assigned to a pool (`!INPOOLS`).

Table 4-2 Initial delivered ACLs setup

Access groups	permissions (access types)	Access pools
AdminGrp	ADMIN ALL	ALL_DEVS
UnknwGrp	BASE	ALL_DEVS
UnknwGrp	ALL	!INPOOLS

Note: The access type `ALL` excludes `ADMIN` privileges.

VLOGIX behavior

During initial setup of the system the access group `UnknwGrp !INPOOLS ALL` is present, as shown in the table above. In this scenario, the `VLOGIX` privilege returns as true since you are granted access to all devices in the Symmetrix access. Initially, since no pools are present the `VLOGIX` privilege is associated implicitly with all devices by this ACL.

Once a user creates an access pool and adds a device to it, the `VLOGIX` privilege is no longer implicitly associated with all devices and, therefore, a check for the `VLOGIX` privilege would now fail. In order for the `UnknwGrp` to still have `VLOGIX` privilege, that privilege must be explicitly granted to the `UnknwGrp` and associated will `ALL_DEVS`.

How to create and manage access groups

Typically various sets of users tend to use the same applications that utilize common Solutions Enabler features from a given host. They typically require the same device resources and permissions of access to these shared devices. For this reason, hosts are registered in groups identified with a group name, which serves as a root for all ACEs in the group (see [Figure 4-1 on page 4-3](#)).

Access groups contain groups of access IDs and their ID names. Any ID and name must belong to just one group and are entered into the database together.

For ease of management, it is highly recommended that you choose an access ID name that best associates with the particular host in use. For example, SunHost1 is more appropriate than a name such as JRSMITH.

Once the group is created, the group name can be used to create access control entries (ACEs).

Setting up an access group

To set up an access control group, you need to create a group and then add the various access IDs of the hosts as members of that group.

Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command:

```
symacl list -v [-sid SymmID|ALL]
```

Create a group

For example, in the command file (**addnewgroups.cmd**), to create access groups named HR and Sales in Symmetrix 12345, enter:

```
create accgroup HR;
create accgroup Sales;
```

You can now commit the file or wait until you have added access IDs to the group. To commit the file (**addnewgroups.cmd**) that executes the creation of the groups, enter:

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

At this point, you may be prompted for your 4 - 12 character access PIN (if environment variable SYMCLI_ACCESS_PIN is not already set to this PIN value).

Add host access IDs to a group

To add a host access ID to an access group, use the following command in the command file:

```
add host accid Id name IdName to accgroup GroupName;
```

For example, in the command file (**addnewgroups.cmd**), to add WinHost with an encrypted ID 73900158-06174491-16225515 to the ProdB group, enter:

```
add host accid 73900158-06174491-16225515 name WinHost to accgroup ProdB;
```

Also in this example, add UXNodeB with encrypted ID 73900158-06174491-16225515 to the group GroupA, enter:

```
add host accid 23300178-56078991-00665410 name UXNodeB to accgroup GroupA;
```

Note: To preserve access ID security for any host, host IDs are encrypted. Use `symacl -unique` to get this encrypted value.

Then you must commit the file (**addnewgroups.cmd**) to execute the addition of these IDs to their groups (and any create group entries if you have not already committed them):

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

Note: After Access Control changes are made, you must run a discover operation (`symcfg discover`) to update your configuration.

User access IDs (PINs) for the AdminGrp

A user access ID is a 4 - 12 character PIN that allows a host to perform commit, prepare, or release operations as the AdminGrp. When a host attempts a commit, prepare, or release operation as the AdminGrp, `symacl` will prompt for this PIN.

To change the user access ID for the AdminGrp access group, use the following command syntax in the command file:

```
add user accid Id name IdName to accgroup AdminGrp;
```

For example, in the command file (**addnewgroups.cmd**), to change the administrative user access ID for JOEPIN to 1234PIN, enter:

```
add user accid 1234PIN name JOEPIN to accgroup AdminGrp;
```

Note: User access IDs may only be set for the AdminGrp access group. Setting an ID for other groups will not return an error, nor prompt for a PIN in most scenarios.

If a host is ever prompted for a PIN, the host should enter 1234PIN. The **addnewgroups.cmd** file must be committed to execute the addition of these new IDs:

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

Editing and managing an access group

Once access groups are established, you can remove access IDs or ACEs from a group, move IDs from one group to another, or delete an entire group.

Remove an access ID from a group

To remove an access ID from an access group, use the following command syntax in the command file:

```
remove accid name IdName from accgroup GroupName;
```

For example, in the command file (**removeaces.cmd**), to remove user HRUser2 from the HR group, enter:

```
remove accid name HRUser2 from accgroup HR;
```

At some point, you must commit the file (**removeaces.cmd**) to execute the removal of this ACE from group HR.

Move an access ID to another group

To move an access ID from one access group to another, use the following command syntax in the command file:

```
move accid name IdName to accgroup GroupName;
```

For example, in the command file (**moveaces.cmd**), to move user HRUser1 to GroupA, enter:

```
move accid name HRUser2 to accgroup GroupA;
```

Then you must commit the file (**moveaces.cmd**) to execute the move of this ACE to GroupA.

Remove all ACEs from a group

To remove all ACEs from an access group, use the following command syntax in the command file:

```
remove aces from accgroup GroupName;
```

For example, in the command file (**removeaces.cmd**), to remove all ACEs from the HR group, enter:

```
remove aces from accgroup HR;
```

At some point, you must commit the file (**removeaces.cmd**) to execute the removal of the ACEs from group HR.

Delete a group

To delete an access group from the database, use the following command syntax in the command file:

```
delete accgroup GroupName [remove_aces=true];
```

Note that when you need to delete a group you either must have removed all ACEs from the group or you can optionally remove all ACEs with the [remove_aces=true] option within this delete command.

For example, in the command file (**deletigroup.cmd**), to delete group HR and any registered ACEs in that group, enter:

```
delete accgroup HR remove_aces=true;
```

At some point, you must commit the file (**deletigroup.cmd**) to execute the removal of access group HR.

How to create and manage limited access to access pools

Access pools are groups of devices controlled by access groups. When the various Symmetrix devices used by a host application must function as non-shareable resources, the target devices must be identified and assigned into a access pool for protection. Once a access pool is created, the pool can be a target to create access control entries (ACEs). More than one access group can access a pool with different permissions. For example, group AdminGrp might access PoolA with ALL permissions, while group HR could access the same PoolA with just BASE permissions.

Note: Once an access pool is created, any host in the access group UnknwGrp is denied access to `symmaskdb list database` command. In addition, when a host in UnknwGrp calls the `symmaskdb list database` command, only it's own devices will be returned in the list. To provide hosts in access group UnknwGrp with full access to the `symmask` commands, grant the access type VLOGIX to access group UnknwGrp. For more information on granting access types, refer to [“Grant permissions to an access group” on page 4-13](#).

Setting up access pools

You can create a access pool and add devices to a pool.

Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command syntax:

```
symacl list -v [-sid SymmID|ALL]
```

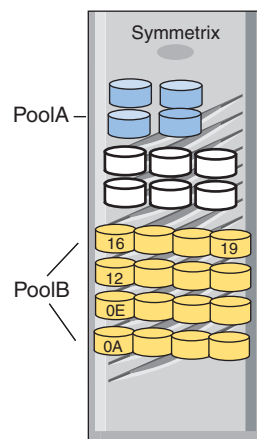


Figure 4-3 Creating access pools

In this example, 16 devices are assigned to PoolB.

Create a pool

For example, in the command file (`addnewpool.cmd`), to create a access pool named PoolB in Symmetrix 12345, enter:

```
create accpool PoolB;
```

You can now commit the file or wait until you have added devices to the pool. To now commit the file (`addnewpool.cmd`) to execute the creation of the pool, enter:

```
symacl -sid 12345 commit -file addnewpool.cmd
```

Add Devices to a Pool

To add the specific Symmetrix devices to a pool, use the following command syntax in the command file:

```
add dev StartDevName[:EndDevName] to accpool PoolName;
```

For example, in the command file (**addnewpool.cmd**), to assign Symmetrix device names 0A through 19 to PoolB, enter:

```
add dev 00A:019 to accpool PoolB;
```

Then you must commit the file (**addnewpool.cmd**) to the database:

```
symacl -sid 12345 commit -file addnewdevs.cmd
```

Editing and managing access pools

You can edit and manage the existing access pool information in the access control database. Devices can be removed from the access pool and access pools can be deleted.

Note: After Access Control changes are made, you must run a discover operation (symcfg discover) to update your configuration.

Remove devices from a pool

To remove specific Symmetrix devices from a pool, use the following command syntax in the command file:

```
remove dev StartDevName[:EndDevName] from accpool PoolName;
```

For example, in the command file (**removedevs.cmd**), to assign Symmetrix device names 16 through 19 for removal in PoolB, enter:

```
remove dev 016:019 from accpool PoolB;
```

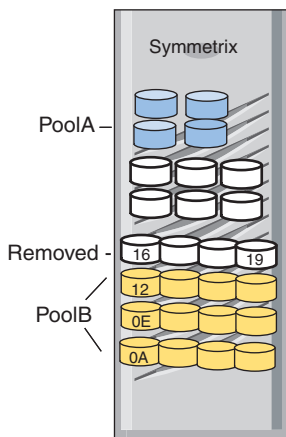


Figure 4-4 Removing devices from a pool

At some point, you must commit the file (**removedevs.cmd**) to execute the removal of devices 16 through 19 from PoolB.

Delete a access pool

You can delete an access pool and optionally at the same time remove any associated access control entries (ACEs) with the specified pool. Note that all ACEs must be removed before you can delete a pool.

To delete an access pool and remove all ACEs, use the following command syntax in the command file:

```
delete accpool PoolName [remove_aces=true];
```

For example, in the command file (**deletepool.cmd**), to remove PoolB and any associated ACEs, enter:

```
delete accpool PoolB remove_aces=true;
```

At some point, you must commit the file (**deletepool.cmd**) to execute the removal of PoolB.

How to create and manage access control entries

Once you have created groups and access pools, you can now create access control entries (ACEs) that grant permissions to the groups and pools.

A group can have multiple permissions and pools, which requires an ACE for each.

Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command syntax:

```
symacl list -v [-sid SymmID|ALL]
```

Grant permissions to an access group

Once you have created an access group, you need to determine and grant, to the access group, the permissions (*AccessType*) of access to a specified pool or to all the devices in the Symmetrix (or all devices not in a pool) using the following command syntax:

```
grant access=<AccessType,,> to accgroup GroupName  
for <accpool PoolName> | <ALL|NON-POOLED devs>;
```

The *AccessType* parameter specifies the permissions to the SYMCLI or SYMAPI features or functionality that is to be granted to selected devices.

Permissions defined

The possible permissions values are listed and described in [Table 4-3](#).

Table 4-3 Permissions: AccessType

Permissions (<i>AccessType</i>)	Description	Commands affected
ADMIN ^b	Grants administrator privilege to grant/deny access control entries to hosts and users.	symacl <prepare, commit, release, list, show> symauth
ADMINRD ^b	Grants read access only to all access control information.	symacl <list, show> symauth <list>
ALL ^b	All possible access types granted except ADMIN and ADMINRD. Must be directed to ALL devices.	All
BASE	Allows the discovery of devices and to obtain states and statistics from the Symmetrix array (directors and devices).	Base component commands symevent, symcfg, symaudit, symld, symdev, symcg <list, show>, symdisk, symstat
BASECTRL	Allows base control operations on devices and device groups.	symld <controls>, symdg <controls>, symcg <controls>
BCV	Allows TimeFinder (BCV) and Clone control and status operations.	symmir <controls> symclone <controls>
CACHCTRL	Allows Cache control operations concerning LRU partition management.	symqos <set LRU >
CFGDEV	Allows powerful configuration control operations that manage various types of configuration changes on devices in the Symmetrix.	symconfigure Allows the following types of operations: • Convert device configurations (BCV, RDF, DRV) • Convert device configuration (changing mirroring) • Set device attributes • Set device emulation • Metadevice management

Table 4-3 Permissions: AccessType (continued)

Permissions (<i>AccessType</i>)	Description	Commands affected
CFGSYM ^b	Allows access to set Symmetrix attributes, set port flags, and swap RA groups with <code>symconfigure</code> . Must be directed to ALL devices.	<code>symconfigure</code> , <code>symmask</code> Allows the following types of operations: • Hot Spare management • SAVE device pool create/delete • SAVE device pool member management • Enable/disable RDFA • Set matrix across Symmetrix array • Add/remove dynamic RDF groups • Change director supporting a dynamic RDF group • Set the link limbo value for a dynamic RDF group
CHECKSUM	Allows Symmetrix device Double Checksum operations.	<code>symchksum <controls></code>
CREATEDV ^b	Allows the creation and deletion of Symmetrix devices (part of <code>symconfigure</code>).	<code>symconfigure</code> • Create new devices • Configure disk space (create/map/mask/meta/attributes) • Delete devices
DIRCTRL ^b	Allows you to take directors and their ports offline and online. Must be directed to ALL devices.	<code>symcfg <online/offline></code> • Online/offline RA directors • Online/offline front end ports <code>symconfigure</code> • Set port flags and attributes
ECC ^{a, b}	Allows the ECC symmetrix agent to run on the requested host.	Not applicable
OPTMZR	Allows user configurable attributes that may affect the Optimizer behavior.	<code>symoptmz <controls></code>
POWRPATH ^{a, b}	Access to PowerPath-directed devices in an RDF consistency group. Must be directed to ALL devices.	Not applicable
QOS	Allows the execution of Quality of Service (QOS) performance control operations to manage copy priorities. Excludes LRU cache control functionality.	<code>symqos <set pace></code>
RCOPY	Manage Open Replicator sessions.	<code>symrcopy</code>
SDDF	Allows the DeltaMark (Change Tracker) functionality that monitors track changes.	<code>symchg</code>

Table 4-3 Permissions: AccessType (continued)

Permissions (<i>AccessType</i>)	Description	Commands affected
SDR	Allows mapping/unmapping of devices to directors/ports for the Symmetrix Disk Reallocation (SDR) feature.	symconfigure - Manage CKD aliases - Map/unmap devices
SNAP	TimeFinder/Snapshot allows the creation and management of virtual copy sessions between a source device and multiple virtual (VDEV) target devices.	symsnap
VLOGIX ^{a, b}	Enables access to Device Masking devices.	symmask, symmaskdb

a. See the appropriate product documentation for use of these access types.

b. These access types must be granted to either ALL devices or all NON-POOLED devices in a Symmetrix.

Before you can grant a group the permissions to any value-added non-base feature, you must first grant the group BASE permissions.

To grant BASE permissions to access group ProdB for all devices, use the following command in the command file (**grantrights.cmd**):

```
grant access=BASE to accgroup ProdB for ALL devs;
```

To grant BASE permissions to access group ProdB for poolPRODdevs, specify:

```
grant access=BASE to accgroup ProdB for accpool PRODdevs;
```

Note: When restricting a host to BASE access to a pool of devices, all devices mapped to the host will be visible along with any devices that are not mapped to the host but are in the pool. To configure a host to only see devices that are in the pool, then only map those devices to the host. In addition, remote (SRDF) arrays and their devices will not be discovered and the Symmetrix application registration database and the audit logs cannot be accessed since these may contain data relevant to other hosts.

To grant RDF permissions to access group ProdB for pool PRODdevs, use the following command in the command file (**grantrights.cmd**):

```
grant access=RDF to accgroup ProdB for accpool PRODdevs;
```

To grant BCV and SDR permissions to access group HR for pool poolHR, use the following command in the command file:

```
grant access=BCV, SDR to accgroup HR for accpool poolHR;
```

Note that these command examples created three ACEs for BASE, BCV, and the SDR permissions, which now could be committed.

To make any non-registered (*unknown*) host have BASE permissions to access group UnknwGrp for all devices in the Symmetrix environment, use the following command in the command file:

```
grant access=BASE to accgroup UnknwGrp for ALL devs;
```

At some point, you must commit the file (**grantrights.cmd**) to execute the granting of permissions with these various command entries.

Remove permissions from an access group

To remove permissions for an access group, use the following command in the command file:

```
remove access=<AccessType,,,> from accgroup GroupName  
for <accpool PoolName> | <ALL|NON-POOLED devs>;
```

To remove SDR permissions from access group HR for poolHR, use the following command in the command file:

```
remove access=SDR from accgroup HR for accpool poolHR;
```

At some point, you must commit the file (**grantrights.cmd**) to execute the change of permissions with these entries.

How to obtain access control information

You can check the current Symmetrix Access Control database to view the various groups, pools, and also the ACLs. Only if you have ADMIN or ADMINRD permissions, can you view all access objects in the access control database.

Using the list action when you don't have administrative permissions, you only see access objects that are associated with the access group to which your host belongs.

Listing access control information

Using the list argument in the following command, you can list information about groups, pools, and ACLs with the following syntax:

```
symacl [-sid SymmID|ALL] [-h]
      list [-v]
      list [-accpool | -accgroup | -acl]
```

For example, the following command lists the access groups on Symmetrix 0133:

```
symacl -sid 0133 list -accgroup
```

The following command example lists all the access pools on Symmetrix 0133:

```
symacl -sid 0133 list -accpool
```

The following example lists all the ACEs (with their pools and access types) for the entire Symmetrix environment:

```
symacl list -acl
```

Showing access control information

Using the show actions when you don't have administrative permissions, you only see access objects that are associated with the access group to which your host belongs.

To show detail information about a specified group or pool, use the following syntax:

```
symacl -sid SymmID|ALL show accgroup GroupName |
      show accpool PoolName
```

For example, the following command shows all the detail about access group ProdB on Symmetrix 0133:

```
symacl -sid 0133 show accgroup ProdB
```

The following command example shows all the detail about the PRODdevs access pool on Symmetrix 0133:

```
symacl -sid 0133 show accpool PRODdevs
```

Getting your host access ID

You can get the access ID for the host you are using. The following command will return a unique access ID in an encrypted format:

```
symacl -unique
```

The `-unique` option returns the encrypted access ID in the following segmented, 24-digit numeric form:

```
xxxxxxxx-yyyyyyyy-zzzzzzzz
```

For example:

```
12301558-94200021-00347892
```

How to release pending locked sessions

During the processing of the command file, the `prepare` and `commit` actions are critical SYMCLI or SYMAPI operations that are considered access control *sessions*. In the event a host machine or application should abnormally fail and stop processing any `prepare` or `commit` access operation on its command file, you can abort this locked session from your current host or another host using the following command:

```
symacl release -sid SymmID
```

Note: If you are the security administrator and you intend to release a lock on the command file session, you must either set environment variable `SYMCLI_ACCESS_PIN` to your access ID or enter your PIN every time `symacl` prompts you.

Verifying a locked session

To verify an access controlled session is locked on any Symmetrix, you can use:

```
symacl list -v
```

This command can tell you how long the session has been locked and who locked it.

Grant/deny access strategies

This section describes the access control strategies that can be applied in an access controlled Symmetrix environment. Several strategies can be considered for establishing or restricting access for a node or group of users or hosts to your Symmetrix environment. These strategies should be considered when you are setting up an access control environment for the first time.

Note: Remember, that after Access Control changes are made, you must run a discover operation (symcfg discover) to update your configuration.

Initially grant all permissions to everybody

The initial (delivered) strategy is to employ a default ID that controls all nodes not yet explicitly registered. This default ID can be used to grant a certain level or a minimal level of access for all unregistered nodes.

In the following example, when the Symmetrix was initially configured (delivered), a group named UnknwGrp was created for non-registered hosts (with no ID).



Figure 4-5 Example default configuration

A special *Default* access ID named `unknown` was added to the group granting all unknown hosts and users ALL permissions.

Next, an ACE was created for group `UnknwGrp` granting them ALL permissions to all the Symmetrix devices. In this scenario, all users and hosts can perform any of the SYMCLI command set operations.

Granting and denying default and legacy access

The delivered setup of Symmetrix Access Control grants legacy applications and hosts whose access IDs have not yet been registered, ALL access to all Symmetrix devices (within the exception of ADMIN and ADMINRD privileges). This was done by creating a group called `UnknwGrp`, and putting the default access ID into it. The name of that default ID was called `unknown`. An access control entry was then granted to the group `UnknwGrp` with a grant of ALL to access all Symmetrix devices.

If this ID is removed from the group `UnknwGrp` or this group is removed, and then you need to add it back to a certain group, use the following syntax:

```
add default accid name IdName to accgroup GroupName;
```

Once the default ID has been added to the group, then all legacy applications and hosts whose access IDs are not yet or never will be registered in the access control database, will default to using the ACEs established for this group.

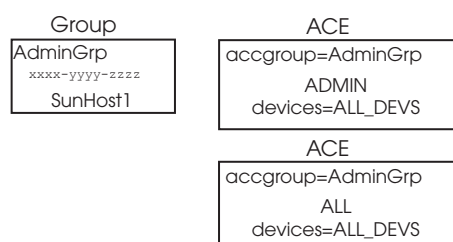
If the default ID is not present, then the access ID of all hosts that need to communicate to the Symmetrix must be registered in the database and granted sufficient permissions to accomplish the functions they need to perform.

Finally, an ACE was created for group UnknwGrp granting them ALL permissions to all the Symmetrix devices. In this scenario, all users and hosts can perform any of the SYMCLI command set operations.

Establishing an administrator

As shown in [Figure 4-1 on page 4-3](#) and [Figure 4-2 on page 4-4](#), the delivered setup allows at least one host assigned with administrative (ADMIN) privileges to be designated the administrator of the information in the access control database.

In the following example, when the Symmetrix was initially configured, an access group named AdminGrp was created. Then a UNIX workstation named SunHost1 with an encrypted access ID (of the form xxxxxxxx-yyyyyyyy-zzzzzzzz) was added to the AdminGrp group. (The access ID was obtained by running `symacl -unique`.)



Next, two ACEs were created; one granting ADMIN permissions and one granting ALL permissions to group AdminGrp for all devices in the Symmetrix.

Grant all permissions to the non-pooled devices

A second possible strategy once access controlled access pools have been established is to allow no restrictions on all the Symmetrix devices that are not otherwise registered (*Non-Pooled*) in any access pool.

In the following example, the ACE for the access group named UnknwGrp was modified to restrict access to only those devices not registered in an access control pool.



General absolute control strategy

A third strategy, which is more controlled, is to only register access to certain devices on an as-needed basis. In this scenario, the UnknwGrp group was removed. Therefore, a node must be known to the Symmetrix, and only specific users/hosts in defined groups with limited or unlimited permissions have access to certain devices defined in their working pools.

Initial setup summary

Whatever strategy you apply during the initial building of the access control database, you need to survey all user needs, organize their limitations into groups, determine what Symmetrix devices to allocate to who, and what permissions are appropriate for each group. It is highly recommended to use the preview and prepare actions on your first major command file before you commit it; particularly if it is an extensive list. The preview and prepare will identify any coding errors or mistakes in the logic.

BCV setup strategy

To set up an access controlled environment for TimeFinder operations, you need set up both the standard and BCV devices as follows:

1. Define your working access pool to contain both the standard and BCV SymDevNames.
2. For your group name, grant BASE permissions to access all devices.
3. For your group name, grant BCV permissions to the access pool holding the pairs.

TimeFinder/Snap setup strategy

To set up an access controlled environment for TimeFinder/Snap operations, you need set up both the source and target devices and the SAVE devices as follows:

Source and target strategy

1. Define your working access pool to contain both the source and target device's SymDevnames.
2. For your group name, grant BASE permissions to access all devices.
3. For your group name, grant SNAP permissions to the access pools holding the pairs.

SAVE device strategy

1. Define your working access pool to contain the SAVE device's.
2. For your group name, grant BASE permissions to access the SAVE devices.
3. For your group name, grant CFGSYM permissions to create and delete the SAVE devices.
4. For your group name, grant SNAP permissions to manage the SAVE devices.

SRDF setup strategy

To set up an access controlled environment for SRDF operations, you need set up both the local Symmetrix and remote Symmetrix. Because both Symmetrix arrays have their own access controlled database, you must:

1. With the ADMIN host, create an access control group for the local Symmetrix. Then with an ADMIN local or remote host, define the same access control group name for the remote Symmetrix.
2. With an ADMIN host, create a access pool defined with the R1 SymDevnames. Then with the ADMIN local or remote host, define a access pool with the R2 SymDevnames.
3. Grant BASE permissions for the group to access all devices. Then with an ADMIN host, grant BASE permissions to access all devices.
4. Grant RDF permissions for the group to access the R1 access pool. Then with an ADMIN host, grant RDF permissions for the group to access the R2 access pool.

How to back up and restore the access control database

When working with access control, it is a good practice to create a backup file of your existing access control database prior to making changes.

Create a backup file

You can create a backup of your current access control database, using the following command:

```
symacl -sid SymmID backup -file CommandFile
```

The backup operation saves the contents of the access control database in the file specified by the file option. The file must not previously exist. The backup file created is compatible for use with the symacl utility.

Note: The backup file contains encrypted versions of the unique IDs, therefore if you compare the values in the backup file to the original file used to create the database, they will be different.

Restore the backup file to the access control database

In the event a host machine or application makes undesirable changes to your access control configuration, you can utilize a backup file you've created to restore the previous configuration data to the access control database.

When you restore a backup using symacl you use the -restore option as shown below:

```
symacl      -sid SymmID commit [-v|-noecho]  
            -restore -file CommandFile
```

The restore operation replaces the contents of the access control database with the contents of the file specified by the file option. It eliminates whatever currently exists in the Symmetrix configuration database and replaces it with the contents of the backup.

Note: The backup file contains encrypted versions of the unique IDs, therefore if you compare the values in the backup file to the original file used to create the database, they will be different.

This chapter describes how to set up user authorization and manage Roles using SYMCLI.

- ◆ User Authorization operations overview5-2
- ◆ User Authorization management.....5-5
- ◆ User Authorization monitoring.....5-8

User Authorization operations overview

User-based authorization provides an additional tool for restricting the management operations that individual users can perform on an array. User-based authorization and Access Control's host-based authorization are independent utilities and none, one, or — for maximum security — both mechanisms can be used.

User authorization is managed using the SYMCLI `symauth` command. Using `symauth`, a user, based upon user name, can be mapped to a specific Role, which defines the operations that they are permitted to perform on the Symmetrix array.

User Authorization is configured independently for each Symmetrix array.

Roles

A Role is a pre-defined set of permissions, or access types, that determine what operations a user can perform. Unlike host-based access control, a user is assigned a particular Role for the entire Symmetrix array not for individual devices (or classes of devices) on it. Roles are predefined in Solutions Enabler and cannot be modified. For each Symmetrix array, a given user can only be assigned a single Role.

There are five Roles defined in Solutions Enabler:

- ◆ **None** — Provides no capability at this time.
- ◆ **Monitor** — Able to perform Read-Only (passive) operations on a Symmetrix array.
- ◆ **StorageAdmin** — Able to perform all management (active, or control) operations on a Symmetrix array.
- ◆ **Admin** — Able to perform all management operations on a Symmetrix array, including security operations.
- ◆ **SecurityAdmin** — Able to perform security operations on a Symmetrix array.

Note: A user cannot change their own role so as to remove Admin or SecurityAdmin privileges from themselves.

The following table defines the permissions assigned to each Role. These permissions are the same as those used in the Access Control `AccessType`. Further details about these permissions can be found in [“Permissions defined” on page 4-13](#).

Table 5-1 Roles defined by permission

Permission/Roles	Monitor	StorageAdmin	Admin	SecurityAdmin
ADMIN			X	X
ADMINRD			X	X
BASE	X	X	X	X
BASECTRL		X	X	
BCV		X	X	
CACHCTRL		X	X	
CFGDEV		X	X	
CFGSYM		X	X	
CHECKSUM		X	X	
CREATEDV		X	X	

Table 5-1 Roles defined by permission (continued)

Permission\Roles	Monitor	StorageAdmin	Admin	SecurityAdmin
DIRCTRL		X	X	
OPTMZR		X	X	
QOS		X	X	
RDF		X	X	
SDDF		X	X	
SDDR		X	X	
SDR		X	X	
SNAP		X	X	
VLOGIX		X	X	
ERASE		X	X	
RCOPY		X	X	

To list the Roles supported by an array, enter:

```
symauth list -roles
```

```
S Y M M E T R I X   A U T H O R I Z A T I O N   R O L E S
```

Role Name	Capabilities
-----	-----
Admin	Authorization and Storage control
SecurityAdmin	Authorization control
StorageAdmin	Storage control
Monitor	View storage environment and status
None	None

Users

Within the user-to-Role mappings, users are identified by User IDs (the `<UserName>` parameter). These IDs consist of a three part string of the form *Type:Qualifier\Name*, where *Type* specifies the type of security authority that was used to authenticate the user, *Qualifier* specifies the specific authority that was used, and *Name* specifies the username relative to that authority. It cannot be greater than 32 characters and spaces are allowed if delimited with quotes.

Authentication types

The following authentication types are supported at this time:

- ◆ **D** — Indicates a user who has been authenticated by either a Windows domain or a Kerberos realm. The qualifier specifies the domain or realm name. For example:
`D:sales\putman` — User putman logged in via a Windows domain sales.
`D:LSS.EMC.COM\jones` — User jones logged in via Kerberos realm LSS.EMC.COM.
- ◆ **H** — Indicate a user who has been authenticated (by logging in) to some host. On Windows, this would correspond to logging into a local account on the host. The qualifier specifies the host name. For example:
`H:jupiter\mason` — User mason logged in on host jupiter.

For a given Symmetrix array, you can assign a user only a single role. Moreover, when you assign this role, it replaces any role currently associated with that user.

UserName format

Within role definition, user IDs can be either fully qualified (as above), partially qualified, or unqualified. When the hostname or domain portion of the `<UserName>` parameter is an asterisk, the asterisk is treated as a wildcard meaning any host or domain. Examples of this include: `H:* \user`, `D:* \user`, and `* \user`. In all other cases, the asterisk is treated as a regular character.

For example, the following are valid `<UserName>` formats:

- ◆ `D:ENG\jones` — Fully qualified path with a domain and user name.
- ◆ `D:* \jones` — Partially qualified that matches user name jones within any domain.
- ◆ `H:\HOST\jones` — Fully qualified path with a host name and user name.
- ◆ `H:* \jones` — Partially qualified that matches user name jones within any host.
- ◆ `jones` — Unqualified user name that matches any jones in any domain on any host.

When a wild card (such as the asterisk) is used, a given user may be matched by more than one mapping in the database. When searching for a user's role, the user authorization mechanism uses the closest ID match that it can find. If an exact match (for example, `D:sales\putman`) is found, that is used, if a partial match (for example, `D:* \putman`) is found, that is used, if an unqualified match (for example, `putman`) is found, that is used, otherwise the user is assigned a role of None.

Note: For any Symmetrix array, a given user can be assigned only a single Role. When a Role is assigned to a user (on some Symmetrix array), it replaces any Role currently associated with that user.

Identify the current user name

To return the current username that Solutions Enabler identifies as accessing the Symmetrix array, enter:

```
symauth show -username
```

```
Your current username is: H:ENG\ljacobs
```

User Authorization management

The `symauth` SYMCLI command is used to manage User Authorization. It allows you to enable and disable User Authorization, set enforcement mode, list defined users and their roles, and assign roles to users.

When working with User Authorization, it is recommend that, prior to enabling User Authorization, you first create your user-to-Role mappings. It is necessary to map at least one user to the Admin or SecurityAdmin role on an array prior to enabling User Authorization.

Enable and disable User Authorization

By default, User Authorization is disabled. As such, any user can make changes to the authorization control data, including creating and removing user-to-Role mappings. Once User Authorization is enabled for an array, only users with the Admin or SecurityAdmin role can change authorization control data.

A user granted the role of Admin or SecurityAdmin can enable or disable User Authorization for a Symmetrix array. To enable User Authorization, enter:

```
symauth -sid 1234 enable
```

Note: Enabling User Authorization is only allowed if the user performing the request has already been granted the Admin or SecurityAdmin role for the Symmetrix array.

Set enforcement

Once User Authorization enabled, failed access attempts can be enforced in one of two ways. An authorization failure can either result in operation failure or simply log a warning. The level of security enforced, can be set using `symauth -sid <SymmID> set enforcement <advise | enforce>`.

- ◆ **advise** — This parameter allows the operation to proceed when authorization is denied, but generates a warning message that gets published to the log file (less secure). This is an effective mode for validating your user-to-Role mapping without preventing access to array functionality. The Solutions Enabler log files and the common audit log will contain an entry of the failed access attempt with the User ID.
- ◆ **enforce** — This parameter, which is the default setting, will cause the operation to fail when authorization is denied (more secure) and log the authorization attempt.

This value can also be set using a command file as described in, [“Command file usage” on page 5-7](#).

Set server policy

For additional security when operating in client/server mode, Solutions Enabler can be set to trust usernames sent from a client (`trust_client`) or require a secure cross-host authentication (`validate_client`), such as Kerberos or Windows. If `validate_client` is set, the server is required to validate a user's identity via a distributed authentication mechanism. For Solutions Enabler version 6.3, validation is only supported in one of the following environments:

- ◆ Client and server are both Windows.
- ◆ Client and server are both Solaris 10, the user has logged in via Kerberos, and the client and server are both part of the same Kerberos realm.

Note: The `validate_client` setting should only be used if the environment meets the criteria stated above. If the server is unable to validate the client's identity, operations performed by the client will fail.

To change the server policy to trust clients, enter:

```
symauth -sid 1234 set server_policy trust_client
```

The `validate_client` option is the default value.

This value can also be set using a command file as described in, [“Command file usage” on page 5-7](#).

Modify user to Role mappings

Users can be assigned a specific role using the `symauth` command by either supplying the user-to-Role mappings via a command file, or, on a UNIX host, by supplying a redirection of STDIN. Each line of the command file must contain a valid username, a supported Role (refer to [“Roles” on page 5-2](#) for supported Roles), and end with a semicolon. For further details on using the command file, refer to [“Command file usage” on page 5-7](#).

To modify user-to-Role mappings via a command file, use the following command syntax:

```
symauth -sid SymmID commit -file <PathName>
```

Where *PathName* is the fully qualified path of the file containing the user-to-Role mappings.

Commit using STDIN

On a UNIX host, users can be assigned a specific role by supplying a redirection of STDIN as shown in the example below:

```
symauth -sid 1234 commit <<!  
assign user H:QA\laura to role Monitor;  
assign user D:Eng\neil to role Admin;  
assign user lauren to role StorageAdmin;  
delete user chris;  
!
```

Command file usage

One or many User Authorization management operations can be added to a command file so a sequence of operations can be previewed and committed at once. A command file is a simple text file of any name and location, the contents of which must be formatted as specified by operation below:

Adding a user to a Role:

```
ASSIGN USER <UserName> TO ROLE <RoleName>;
```

Reassigning an existing user to Role:

```
REASSIGN USER <UserName> TO ROLE <RoleName>;
```

Deleting a user:

```
DELETE USER <UserName>;
```

Set enforcement:

```
SET ENFORCEMENT [ADVISE | ENFORCE];
```

Set server policy:

```
SET SERVER_POLICY [TRUST_CLIENT | VALIDATE_CLIENT];
```

Where:

- ◆ *UserName* — The name of a user, (32 character maximum). Spaces are allowed if the <UserName> is quote delimited. Examples of a valid <UserName> include: "D:domain\joe", "H:host\joe", "domain\joe", and "joe". In the case of "domain\joe", the <UserName> is interpreted as D:domain\joe. In the case of joe, this <UserName> is allowed regardless of domain or host.

Note: When the hostname or domain portion of a <UserName> is an asterisk, the asterisk is treated as a wildcard meaning any host. Examples of this include: H:*user, D:*user, and *user. In all other cases, the asterisk is treated as a regular character.

- ◆ *RoleName* — The name of the role to be assigned to a user. Current valid roles include: Admin, SecurityAdmin, StorageAdmin, Monitor, and None. These are case insensitive.

Preview and Commit actions

To safely apply any of these authorization actions (command file entries) to the array, you should perform the following progressive *symauth* operations on the command file:

- ◆ **Preview** — *symauth -sid SymmID preview -file <PathName>*
After you first create the command file, the *preview* operation verifies the syntax and correctness of the contents of the entries in the command file.
- ◆ **Commit** — *symauth -sid SymmID commit -file <PathName>*
The *commit* operation performs both the preview and prepare checks and then commits the contents of the command file to the Symmetrix array.

Note: It is not mandatory to execute a preview action prior to a commit. However, a preview action can ensure that the commit action will not be rejected or can be used to debug the command file entries.

User Authorization monitoring

Monitoring operations allow you to view the current User Authorization settings on a specific array.

View User Authorization policies

To list the User Authorization settings (policies) in effect, enter:

```
symauth -sid 1234 list
```

```
Symmetrix ID: 000000001234
```

```
Authorization Control : Enabled
```

```
Time Enabled          : Thu Jun  8 15:02:14 2006
```

```
Time Disabled         : Thu Jun  8 12:01:44 2006
```

```
Time Updated          : Thu Jun  8 15:02:14 2006
```

```
Authorization Mode     : Advise
```

```
Server Policy          : Trust clients
```

View user-to-Role mappings

List the user-to-Role mappings currently established.

```
symauth -sid 1234 list -users
```

```
Symmetrix ID: 000000001234
```

```
Role name              Username
```

```
-----
```

```
Admin                  moe
```

```
Admin                  D:Eng\moe
```

```
StorageAdmin          larry
```

```
Monitor                curly
```

Additional options can be specified to sort the return list by Role (-by_role), user (-by_user), or domain(-by_domain).

View Roles supported

To list the Roles supported by an array, enter:

```
symauth list -roles
```

```
S Y M M E T R I X   A U T H O R I Z A T I O N   R O L E S
```

```
Role Name              Capabilities
```

```
-----
```

```
Admin                  Authorization and Storage control
```

```
SecurityAdmin          Authorization control
```

```
StorageAdmin          Storage control
```

```
Monitor                View storage environment and status
```

```
None                   None
```

This chapter describes the benefits of grouping devices, the types of groups, and how to create and modify them.

◆ Overview.....	6-2
◆ How to create a device group	6-5
◆ Adding devices to a device group.....	6-6
◆ Device groups: Exporting and importing device lists.....	6-11
◆ Renaming, moving, and deleting device groups	6-13
◆ Composite groups	6-16

Overview

A collection of devices can be assigned to a named group to provide a more manageable object to query status and impart control operations. Groups can be used to identify and work with a subset of available Symmetrix devices, obtain configuration, status, and performance statistics on a collection of related devices, or issue control operations that apply to all devices in the specified device group.

Grouping types

Devices can be associated as either a device group or a composite group.

- ◆ A *device group* (DG) is a user-defined group comprised of devices that belong to a single Symmetrix array and a single RDF (RA) group. A control operation can be performed on the group as a whole, or on the individual device pairs that comprise it. By default, a device cannot belong to more than one device group and all of the STD devices in a group must reside on the same Symmetrix array. However, if the Symmetrix options file parameter SYMAPI_ALLOW_DEV_IN_MULT_GRP is enabled, a device can be added to multiple groups. You can use device groups to identify and work with a subset of available Symmetrix devices, obtain configuration, status, and performance statistics on a collection of related devices, or issue control operations that apply to all devices in the specified device group.
- ◆ A *composite group* (CG) is a user-defined group comprised of devices that can belong to one or more locally-attached Symmetrix arrays and one or more RDF (RA) groups within a Symmetrix. For more information on composite groups, refer to [“Composite groups” on page 6-16](#).
 - An *RDF consistency group* is a CG comprised of RDF devices (RDF1 or RDF2), which has been enabled for RDF consistency. The RDF consistency group acts in unison to preserve dependent write consistency of a database distributed across multiple SRDF systems. It maintains this consistency via PowerPath or Multi Session Consistency (MSC), which respects the logical relationships between dependant I/O cycles. Refer to [“What is an RDF consistency group?” on page 6-16](#) for information on using RDF consistency groups.

Group name services (GNS)

In a default Symmetrix environment, device group and composite group definitions are created via a locally attached host. Upon creation, the group definition is stored in the host's configuration database file. Therefore, only the host that created the group would see the group and control it. To perform control operations from another locally attached host, the group definition would have to be manually copied to other hosts.

Optionally, you could enable the *Group Name Services (GNS)* on your Symmetrix-based host's. GNS provides the optional ability to store device and composite group definitions in a shared repository located on each Symmetrix array, which would then automatically become visible to all locally attached hosts. This allows all GNS-enabled hosts to see the same group definitions across your Symmetrix environment, while sharing real-time updates to group definitions and configurations made by other hosts.

For more information on GNS, refer to [Chapter 7, “Group Name Services.”](#)

Group names and device names

Device groups, as well as the devices in a device group, are assigned names that facilitate reference in a session. You assign a device group name at the time you create it. The name can have up to 31 characters and must be unique for a given configuration database.

When you add a device to a device group, it is given a logical name. This name allows you to refer to the device independently of its physical device name or Symmetrix device name. The name can have up to 31 characters and must be unique within the device group. It is known only within the context of the device group to which the device belongs. You must use this logical name in any SYMCLI command that has the argument `LdevName` or `ld`.

Device group types

The Solutions Enabler Base product manages three types of device groups: REGULAR (non-RDF), RDF1 (RDF source device), and RDF2 (RDF target device). When you create a device group in an RDF configuration, you must specify the type of the device group (RDF1 or RDF2). Otherwise, the group type defaults to REGULAR when no type is specified. The following device lists can be maintained in device groups:

- ◆ **SRDF device list** — You can group SRDF (Symmetrix Remote Data Facility) standard devices in a device group subject to the following restrictions:
 - You cannot mix SRDF and non-SRDF devices in the same device group.
 - All SRDF devices in a given device group must be either all source (R1) or all target (R2) devices.
 - All SRDF devices in a given device group must belong to the same Symmetrix RDF group¹ or if concurrent RDE, then two RA groups.

All devices on a Symmetrix that are remotely mirrored using SRDF need not be remotely mirrored to the same Symmetrix. Some RDF devices on a Symmetrix can be mirrored to one Symmetrix while other RDF devices can be mirrored to another Symmetrix.

- ◆ **TimeFinder/BCV device list** — You can associate Business Continuity Volume (BCV) regular devices and RDF1 BCV devices with a device group. You can do BCV control operations on any BCV pair in the device group. Also, you can do RDF operations on just the associated RDF1 BCV devices. Refer to *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide* for more information on adding BCV devices to a device group.
- ◆ **TimeFinder/Snap virtual device list** — A virtual device is a host-accessible device containing address pointers to the data stored on the source device or a pool of save devices, which indicate where the copy session data is located in the physical storage. You can associate virtual devices (VDEV) paired with standard and BCV devices with any device group. In addition, TimeFinder/Snap control operations can be performed on any virtual device in a device group.
- ◆ **TimeFinder/Clone target list** — Target lists provides a mechanism establishing source (SRC) and target (TGT) devices for TimeFinder/Clone operations. They can be created for both device groups and composite groups. A Target list can contain various types of

1. RDF groups (also known as RA groups) are configured by the Symmetrix Enginuity software. For Enginuity 5x67 or higher, when concurrent RDF devices are configured, any R1 device may be mapped to two R2 devices, each one belonging to a different RA. R2 devices belong to one RA group. Otherwise, an RA director and any one SRDF storage device can belong to just one RDF group. Typically, there are two or more RA directors per RDF group. The Enginuity load balances I/O on all healthy RA directors in an RDF group.

devices, including STDs or BCV devices (based on a set of rules discussed in [“Clone target restrictions” on page 6-5](#)) and use those devices as targets in Clone operations. Remote Target lists can also be created for remote operations.

- ◆ **Gatekeeper device list** — You can *associate* one or more gatekeeper devices with a device group. SYMCLI will use the associated gatekeeper to issue requests to the Symmetrix for control operations on the devices within the specified device group. You can *add* a standard device to a device group. However, the gatekeeper cannot be added to the device group, only associated with a device group. For more information about associating a gatekeeper device with a group, refer to [Chapter 10, “Gatekeeper Devices.”](#)

Note: A BCV device or a RDF2 device cannot be assigned as a gatekeeper, nor can a device that is a member of a device group be defined as a gatekeeper.

How to create a device group

Device group is created by first defining a named empty group of a specific type. Then, devices are added to the group. A newly created device group is defined by the devices you add to it and each type of device list has its own set of restrictions.

Creating an empty device group

Device groups are created using the `syndg` command. When you create a device group, you assign it a name and choose its type: REGULAR (non-RDF), RDF1 (RDF source device), and RDF2 (RDF target device). If a type is not specified, the default group created is REGULAR.

For example, to create a device group named `prod` whose members are operating as SRDF source devices, enter:

```
syndg -type RDF1 create prod
```

Device restrictions

The following are some basic restrictions on the devices that can be added to a device group based upon device type:

SRDF device restrictions

If you are adding an SRDF device to a device group, all devices in the device group:

- ◆ Must be SRDF devices.
- ◆ Must be either all source (RDF1 type) or all target (RDF2 type) devices.
- ◆ Must have the same Symmetrix RDF group number. The only exception to this rule is in a concurrent RDF configuration where the R1 has two R2s on two separate arrays.

Note: RDF groups (also known as RA groups) are configured and assigned RDF group numbers (1 to 64) by the Symmetrix Enginuity software. An RA director and any one storage device in the SRDF can belong to only one RDF group. Typically there are two or more RA directors per RDF group. RDF groups can be seen by entering: `symcfg list -RA all`

Virtual device restrictions

To add a device as a VDEV, the device must be defined as a virtual device.

Clone target restrictions

Both STD or BCV devices can be added to a target list. Below are the sets of device types allowed in a target list, though devices from only one set are allowed in a given device group's target list at any given time:

- ◆ Non-RDF STDs
- ◆ R1 STDs
- ◆ R2 STDs
- ◆ R1 + Non-RDF Standards
- ◆ R2 + Non-RDF Standards
- ◆ Non-RDF BCVs.
- ◆ R1-BCVs
- ◆ R2-BCVs
- ◆ R1-BCVs + Non-RDF BCVs
- ◆ R2-BCVs + Non-RDF BCVs

By default, the logical device name (LdevName) for devices in a target list will be TGTxxx. For devices in Remote Target List, default Ldevname will be RTGTxxx.

Adding devices to a device group

SYMCLI provides various commands to add standard devices (STD), virtual devices (VDEV or RVDEV), or TimeFinder/Clone target (TGT or RTGT) devices to a device group. Once a device group has been created, SYMCLI commands can be used to add a single device (`symld`), all devices on a Symmetrix not yet assigned to a device group (`symld`), or a list of devices from a file (using `symdg`) to that group. For more information on adding a list of devices from a file, refer to [“Device groups: Exporting and importing device lists” on page 6-11](#).

Add devices by physical device name

The `symld` command is used to add a single device to a device group. All devices in a device group must be part of the same Symmetrix array. Devices can be added by specifying either the physical device name (`add pd`) or the Symmetrix device name (`add dev`). When adding virtual devices or target devices specify the Symmetrix device name (`SymDevName`). The following is the syntax used to add a device:

```
symld -g DgName [-h] [-offline]

add pd PdevName [LdevName]

add dev SymDevName [LdevName] [-sid SymmID]
[<-vdev | -tgt> -rdf [-rdfg GrpNum]]
```

To add a single device using the physical device name `/dev/rhdisk32` to a device group named `prod`, enter:

```
symld -g prod add pd /dev/rhdisk32
```

Optionally, you can assign a logical device name to the device. This logical device name can have up to 31 characters and must be unique within the device group. If a logical device name is not specified, one will be supplied automatically by SYMCLI.

To add Symmetrix device `0005` to a device group named `prod` and assign the logical device name `temp1`, enter:

```
symld -g prod add dev 0005 temp1
```

Adding virtual devices

You can add a virtual device to the virtual device list (VDEV) of a device group or the remote virtual device list (RVDEV). When adding virtual devices from a local array, specify the `-vdev` option along with the Symmetrix device name. For example, to add virtual device `0005` to a device group named `prod1` and assign the logical device name `vdev1`, enter:

```
symld -g prod1 add dev 0005 vdev1 -vdev
```

When adding virtual devices from a remote array (RVDEV), specify the `-vdev -rdf` option along with the Symmetrix device name. This targets the operation to the specified virtual device over RDF links on the remote array. In the situation where you have concurrent RDF, whereby there are two Symmetrix arrays on the remote side, you must specify the RDF group number (`-rdfg GrpNum`) with `-rdf` option.

For example, to add virtual device `0006` belonging to RDF group `12` to a device group named `prod2` and assign the logical device name `vdev2`, enter:

```
symld -g prod2 add dev 0006 vdev2 -vdev -rdf -rdfg 12
```

Adding devices to the target list

You can add a device to the target device list (TGT) of a device group or the remote target device list (RTGT) for TimeFinder/Clone operations. When adding devices from a local array, specify the `-tgt` option along with the Symmetrix device name. For example, to add target device 0023 to a device group named `prod1` and assign the logical device name `tgt23`, enter:

```
symld -g prod1 add dev 0023 tgt23 -tgt
```

When adding virtual devices from a remote array, specify the `-tgt -rdf` option along with the Symmetrix device name. This targets the operation to the specified target device over RDF links on the remote array. In the situation where you have concurrent RDF, whereby there are two Symmetrix arrays on the remote side, you must specify the RDF group number (`-rdfg GrpNum`) with `-rdf` option.

For example, to add target device 0069 belonging to RDF group 12 to a device group named `mywork` and assign the logical device name `tgt2`, enter:

```
symld -g mywork add dev 0069 tgt2 -vdev -rdf -rdfg 12
```

For details on the types of devices that can be added to a device group's target list, refer to ["Clone target restrictions" on page 6-5](#).

Adding all ungrouped devices

Use `symlld` to add all un-grouped devices on a Symmetrix array to a device group. By default, this command will add all standard devices (or local virtual devices) to a device group. However, various optional criteria can be set to specify the types of devices to add to the specified device group. You can specify that the following devices are added to a device group:

```
symlld -g DgName [-h] [-offline]
```

```
addall [-sid SymmID] [-SA <#|ALL>] [-P #]
        [<-vdev | -tgt> -rdf [-rdfg GrpNum]] [-CAP #] [-N #]
        [-RANGE <start>:<end>] [pd | dev | -host HostName]
```

- ◆ All un-grouped physical (pd) or Symmetrix (dev) devices from a specific Symmetrix ID (-sid)
- ◆ All devices visible to a specified host's front-end directors (-host *hostname*)
- ◆ All devices visible to one or all front-end directors (-SA <#|ALL>)
- ◆ All devices visible to one or all front-end port number (-P #)
- ◆ All devices that exist within a specified range (-RANGE <start>:<end>), or that are of a specific capacity (-CAP #)
- ◆ All virtual devices form a local or remote array (<-vdev> [-rdf [-rdfg GrpNum]])
- ◆ All devices are to be added to the target list of the device group for TimeFinder/Clone operations (<-tgt> [-rdf [-rdfg GrpNum]])

Note: In the situation where you have concurrent RDF, whereby there are two Symmetrix arrays on the remote side, you must specify the RDF group number (-rdfg *GrpNum*) with -rdf option.

For example, to add all existing physical devices visible to your host within Symmetrix 063 to device group `prod`, enter:

```
symlld -g prod -sid 063 addall pd
```

The `addall` argument assigns the following logical names to the devices it adds: DEV001, DEV002,..., DEV*nnn*. You can use the `rename` argument to change these logical names after the devices have been added. Or, prior to calling this command entry, you can change the default logical device naming conventions using the `SYMCLI_LDEV_NAMING` environment variable. For more information on setting environment variables, refer to *EMC Solutions Enabler Symmetrix CLI Command Reference*.

When using the `addall` you **cannot**:

- ◆ Mix devices from different Symmetrix arrays
- ◆ Mix RDF devices having different RDF group numbers
- ◆ Add device(s) defined as a gatekeeper or BCVs
- ◆ Add devices whose device type does not match the device group type

Listing device groups

To retrieve a list of all device groups defined in your configuration database, enter:

```
symdg list
```

An example output appears below:

D E V I C E G R O U P S								
Name	Type	Valid	Symmetrix ID	Devs	GKs	Number of		TGTs
						BCVs	VDEVs	
Team1A	RDF1	Yes	000000006164	2	0	0	0	2
Eng1	REGULAR	Yes	000187900039	14	0	5	0	4
Acct_Team23	RDF1	Yes	000187900039	1	0	0	0	1
All	REGULAR	Yes	000187940256	1	0	16	2	1

This command returns a list of all device group names, group type, Symmetrix ID, and the number of standard, BCV, gatekeeper, virtual devices, and TimeFinder/Clone target devices (TGTs).

Show details about a device group

Use `symdg` to gather detailed information about a specific device group. For example, to show information about device group `clonetgtdg`, enter:

```
symdg show clonetgtdg
```

Note: The example output below has been truncated and is intended to provide the overall example of a possible data set.

```
Group Name: clonetgtdg
Group Type           : RDF2      (RDFA)
Device Group in GNS  : Yes
Valid                : Yes
Symmetrix ID         : 000000006164
Group Creation Time  : Fri Nov 18 15:21:27 2005
Vendor ID            : EMC Corp
Application ID       : SYMCLI
Number of STD Devices in Group : 1
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 0
Number of Locally-associated VDEV's : 0
Number of Locally-associated TGT's : 1
Number of Remotely-associated VDEV's (STD RDF) : 0
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 0
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 0
Number of Remotely-associated TGT's (TGT-RDF) : 1
Standard (STD) Devices (1):
{
-----
LdevName          PdevName          Sym      Cap
Dev   Att.  Sts      (MB)
-----
DEV001            N/A              0160      WD      1031
}
Target Devices Locally-associated (1):
{
-----
LdevName          PdevName          Sym      Cap
Dev   Att.  Sts      (MB)
-----
```

```
-----
TGT001          N/A          016A      WD      1031
}
Target Devices Remotely-associated (1):
{
-----
LdevName          PdevName          Sym          Cap
Dev  Att.  Sts          (MB)
-----
RTGT001          N/A          026A      WD      1031
}
```

Device groups: Exporting and importing device lists

You can save the list of devices from an existing group to a file on your host system, and this file can later be imported to create a device group. This can be useful if you then delete the group and later wish to recreate it, or if you wish to import the group definition to another system.

Exporting a device list

You can remove all devices from a group but retain or export the list of devices in the group to a file on your host system using the following syntax:

```
symdbg
  export DgName [-delete] [-f FileName]
    [[-rdf [-rdfg GrpNum]] | [-sid SymmID]]

  exportall [-delete] [-f FileName]
    [[-rdf [-rdfg GrpNum]] | [-sid SymmID]]
```

For example, to export the device group membership from group `prod2` to file `prod2list`, enter:

```
symdbg export prod2 -f prod2list
```

To export an RDF group, specifying the `-rdf` option will use the remote Symmetrix ID and device names and change the RDF group type from R1 to R2 or vice versa. In addition, an RDF group number can be specified as well (`-rdfg GrpNum`).

In addition, a text file that contains the details of all members of the existing device groups can be created using the `symdbg exportall` operation. The device groups can later be recreated from this file using the `symdbg importall` command.

Exporting and deleting a device group

The delete option allows you to export the device group membership to a file and delete the existing device group in the same operation. To reinstate the same group again, you can import this list (refer to [“Importing device lists” on page 6-12](#)) to the same or different device group.

Exporting device lists to OSM environments

Using the `dg2file` action, you can create a Symmetrix Manager (OSM) device file from the contents of a device group.

For example, to create a device file named `prod_OSMlist` that lists existing standard/BCV pairings in device group `prod3`, enter:

```
symdbg dg2file prod3 -f prod_OSMlist -ftype STD_BCV
```

Exporting device lists with -delete option

You can delete a group but export the list of devices in the group to a file on your host system.

For example, to export the device group membership to file `prod2list` from group `prod2` and then delete the group, enter:

```
symdbg export prod2 -f prod2list -delete
```

For more information on deleting a device group, refer to [“Deleting a device group” on page 6-15](#).

Importing device lists

You can add multiple devices to a new or existing device group by importing an existing file that contains a list of devices created by the export action.

For example, to create a device group named `prod2` from the file `prod2list`, enter:

```
symdg import prod2 -f prod2list
```

Typically, you import files that were previously exported (refer to [“Exporting a device list” on page 6-11](#)).

The `import` action will create the device group if the group name specified in the command does not already exist, or you can import to an existing group name that is partially populated. If you import to an existing group, the devices in the imported file will be appended to the existing group membership.

In addition, you can recreate all device groups, via the `symdg importall` command, that were from data contained in a text file previously created using the `symdg exportall` command.

Importing OSM lists

Using the `file2dg` action, you can also create a device group by importing device files that were formatted by the Symmetrix Manager (OSM) utility.

For example, to create a device group named `prod3` from device file `prod_OSMlist`, enter:

```
symdg file2dg prod3 -f prod_OSMlist
```

Renaming, moving, and deleting device groups

Solutions Enabler also provides the ability to rename, move, and delete existing device groups.

Renaming

Device groups, and the logical device name of devices in a device group, can be renamed.

Renaming a device group

Use `symsg` to rename a device group. The new name can contain up to 31 characters and must be unique for the configuration database on which the device group is defined.

For example, to rename the device group `prod` to `prod_B`, enter:

```
symsg rename prod prod_b
```

Renaming devices

Use `symld` to change the logical name of a device. The name can have up to 31 characters and must be unique within its device group.

For example, to rename the logical name of device `DEV003` to `TEMP3` in device group named `prod`, enter:

```
symld -g prod rename DEV003 TEMP3
```

Moving devices

Use `symld` to move one device, specifying the logical device name, from one existing device group to another existing device group. The source and destination device groups must be compatible types. The following syntax is used to move an individual device:

```
symld -g DgName [-h] [-offline]  
  
move LdevName DestDgName [-force] [-rename]
```

For example, to move logical device `DEV003` from device group `prod` to device group `test`, enter:

```
symld -g prod move DEV003 test
```

If there exists a device within the destination device group that has the same logical device name as device you wish to move into the destination device group, you must use the `-rename` option to avoid encountering an error. When this option is used, SYMCLI will rename the moved device to the next available logical device name as defined in the `SYMCLI_LDEV_NAMING` environment variable.

Moving all devices

When choosing to move all devices from one existing device group to another, you can move all devices, or use the additional options to focus the operation to devices that meet a specific criteria. By default, this operation only affects standard devices. However, the other option can be specified to only move specific device types as shown in the following syntax:

```
symld -g DgName [-h] [-offline]  
  
moveall DestDgName [-force] [-rename]  
    [-SA <#|ALL>] [-P #] [-CAP #]  
    [-N #] [-RANGE <start>:<end>]  
    [-vdev | -rvdev | -tgt | -rtgt]
```

Options include specifying devices mapped to a specific front-end (SCSI or Fibre) director number, (`-SA <#|ALL>`), devices mapped to a specific (SCSI or Fibre) director port number

(-P #), devices of a specified capacity (-CAP #), total number of devices (-N #), or a range of devices (-RANGE <start>:<end>) to move. The source and destination device groups must have compatible types.

For example, to move all virtual devices from device group `prod` to device group `test`, enter:

```
symld -g prod moveall -vdev test
```

Copy devices from a DG to a CG

Devices from an existing device group can be copied (or added) to an existing composite group. Use the following syntax to add devices from an existing device group to an existing composite group:

```
symdg dg2cg DgName CgName
[-bcv|-vdev|-rbcv|-rrbcv|-brbcv|-nobcv|-rvdev|-tgt|-rtgt]
[-apidb | -ppath | -rdf_consistency] [-force] [-rename]
```

The following additional options are available:

- ◆ -bcv adds only the BCVs to the composite group.
- ◆ -vdev adds only the Symmetrix virtual devices to the composite group.
- ◆ -rbcv adds only the RBCVs to the composite group.
- ◆ -rrbcv adds only the RRBCVs to the composite group.
- ◆ -brbcv adds only the BRBCVs to the composite group.
- ◆ -nobcv adds only the STDs to the composite group.
- ◆ -rvdev adds only the Symmetrix remote virtual devices to the composite group.
- ◆ -tgt adds only the TGTs to the composite group.
- ◆ -rtgt adds only the RTGTs to the composite group.

By default, all device lists from the device group are added to the composite group.

Setting consistency

The [-rvdev | -apidb | -ppath | -rdf_consistency] options allow you to choose a consistency method for the CG. If the -rvdev parameter is specified then Symmetrix remote virtual devices will be added. If the -apidb parameter is specified then the composite group will only be committed to the SYMAPI configuration database and consistency is not maintained. If the -ppath parameter is specified then the composite group will be created in PowerPath as well, and PowerPath will maintain consistency. If -rdf_consistency is specified it allows it to be enabled for RDF consistency once devices have been added to the CG. RDF consistency protection is supported by the RDF daemon, which currently supports SRDF/A Multi Session Consistency (MSC).

Preventing name collision

The -rename option allows SYMAPI to use a unique device group name on all hosts, preventing any name collision that would cause the operation to fail.

Removing devices

Use `symld` to remove a device from a device group. Devices can be removed individually using the `remove` argument, or all devices can be removed from a device group using `rml1`. By default, the `remove` and `rml1` arguments will affect standard devices only. However, the `-vdev` option can be specified to remove just the virtual devices.

For example, to remove logical device `DEV003` from a device group named `prod`, enter:

```
symld -g prod -force remove DEV003
```

In this example, the `-force` option was used to remove the device regardless of its BCV state.

Note: If you remove the only member of a device group, the device group is not automatically deleted. You must execute `symdg delete` to explicitly delete the device group, as described in the [“Deleting a device group” on page 6-15](#).

Removing all devices

When choosing to remove all devices from an existing device group, you can use the additional options shown in the following syntax to focus the operation to all devices that meet a specific criteria:

```
symld -g DgName [-h] [-offline]
      rml1 [-force] [-SA <#|ALL>] [-P #] [-CAP #] [-N #]
      [-RANGE <start>:<end>]
      [<-vdev | -tgt> -rdf [-rdfg GrpNum]]
```

Options include specifying devices mapped to a specific front-end (SCSI or Fibre) director number, (`-SA <#|ALL>`), devices mapped to a specific (SCSI or Fibre) director port number (`-P #`), devices of a specified capacity (`-CAP #`), total number of devices (`-N #`), a range of devices (`-RANGE <start>:<end>`) to remove, or a local and remote virtual and target devices (`[<-vdev | -tgt> -rdf [-rdfg GrpNum]]`).

Deleting a device group

Use the following syntax to delete a device group:

```
symdg
      delete DgName [-force]
```

For example, to delete device group named `prod`, enter:

```
symdg delete prod
```

Note: Deleting populated device groups requires the use of the `-force` option.

Composite groups

A Composite Group (CG) is a user-defined group whose STD device members can belong to multiple Symmetrix arrays and RA groups.

A single composite group can contain devices from the following different device lists:

- ◆ Standard device list (STD) — Non-BCV devices that are local to the host.
- ◆ Local Business Continuance Volume (BCV) list — BCV devices local to the host.
- ◆ Local VDEV list (VDEV) — Virtual devices that are local to the host.
- ◆ Remote VDEV list (RVDEV) — Virtual devices that are remote.
- ◆ Remote BCV list (RBCV) — BCV devices that are to be associated with the remote mirrors of the STD devices.
- ◆ BCV-Remote BCV list (BRBCV) — BCV devices that are to be associated with the remote mirrors of the local BCV devices.
- ◆ Remote-Remote BCV List (RRBCV) — Remote BCV devices that are to be associated with the remote mirrors of the RBCV devices.
- ◆ Local TGT list (TGT) — TimeFinder/Clone target devices that are local to the host.
- ◆ Remote TGT list (RTGT) — TimeFinder/Clone target devices that are remote.

A composite group can be defined as type REGULAR, RDF1, or RDF2.

What is an RDF consistency group?

A RDF consistency group is a composite group comprised of RDF devices (RDF1 or RDF2 devices) acting in unison to preserve dependent write consistency of a database distributed across multiple SRDF systems. It maintains this consistency by using either Multi Session Consistency (MSC) or PowerPath.

A group is considered a RDF consistency group if it is a composite group that meets ALL of the following criteria:

- ◆ Created as type RDF1 or RDF2
- ◆ Marked as being in PowerPath or register with the RDF daemon
- ◆ Contains STD devices
- ◆ Enabled for consistency

Note: For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at <http://elabnavigator.emc.com>.

Multi session consistency (MSC)

If a source R1 device in the consistency group cannot propagate data to its corresponding target R2 device, SRDF/A MSC suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets.

SRDF/A MSC cycle-switching and cache recovery operations across all SRDF/A sessions in the consistency group ensures a consistent R2 copy of the database up to the point in time that the interruption occurred. When an interruption occurs, SRDF/A MSC examines the status of the SRDF/A sessions and data. Based on the results, MSC either commits the last copy cycle to the R2 side or discards it.

Achieving data consistency across multiple RDF (RA) groups requires that the cycle switch process be coordinated among participating Symmetrix arrays, and that the switch occur during a very brief period (usually a few milliseconds) when no host writes are being serviced. SRDF/A MSC provides a single coordination point to drive the cycle switch process in all participating Symmetrix arrays.

PowerPath consistency

If one or more source (R1) devices in a RDF consistency group cannot propagate data to their corresponding target (R2) devices, the PowerPath-connected RDF consistency group devices suspend all the source (R1) propagation from these devices, as shown in [Figure 6-1 on page 6-18](#).

This action instantly and completely halts data flow to the consistency group's target (R2) side ensuring that a consistent database exists (up to the point-in-time of data propagation failure) on the remote side of the configuration, and preserving the integrity of the remote database.

I/O to the local consistency group's devices automatically resumes when the consistency group's RDF source (R1) are suspended. While these updates are not immediately sent to the target side, they are propagated through normal SRDF operation once SRDF mirroring is resumed. RDF consistency group operations and can be managed using the `symcgr` command.

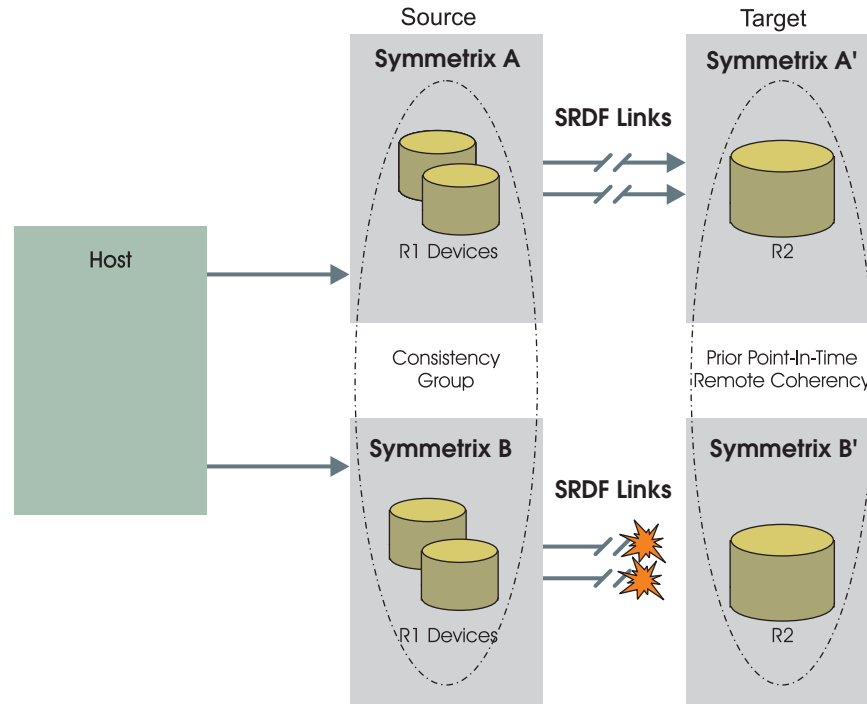


Figure 6-1 RDF consistency groups

PowerPath and composite groups

PowerPath is not required to create and use composite groups. In fact, by default, newly created composite groups of type RDF1 or RDF2 are not automatically added to PowerPath even if a valid PowerPath version is installed on the host. However, consistency enabled composite groups can be added to PowerPath by configuring global behavior parameters, SYMAPI_RDF_CG_TO_PPATH in the SYMAPI options file, for global changes or by utilizing the command-line option `-ppath` (PowerPath only). For more information on enabling the SYMAPI_RDF_CG_TO_PPATH option, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Note: You must use the default SYMAPI configuration database to use RDF consistency groups with PowerPath.

Creating composite groups

The `symcg` command is used to create and manage composite groups. Use the following command syntax to create a composite group of type REGULAR, RDF1 or RDF2:

```
symcg create CgName [-type REGULAR | RDF1 | RDF2]
                  [-apidb | -ppath | -rdf_consistency]
```

If you do not specify a type during the `symcg create` operation, the group will be set as a REGULAR group by default.

If `-apidb` is specified the composite group will not be added to PowerPath (for more information, refer to [“Creating composite groups without consistency” on page 6-19](#)). When creating a composite group of type RDF1 or RDF2, when the `-ppath` option is specified the composite group will be added to PowerPath (for more information, refer to [“Creating composite groups with PowerPath” on page 6-19](#)). When creating a composite group of type RDF1 or RDF2, when the `-rdf_consistency` option is specified the composite group will be

added to RDF daemon. For more information, refer to [“Creating composite groups with PowerPath”](#).

Note: When a composite group is created as type RDF1 or RDF2 and the added to PowerPath (either by specifying the `-ppath` option or by configuring the `SYMAPI_RDF_CG_TO_PPATH` option file setting, assuming no other command-line option is specified), the type is initially ignored. On subsequent list and show commands it will be shown as NA until the first RDF STD device is added to the composite group when it will revert to the created type.

Creating composite groups with PowerPath

When a composite group is created and it is deemed to be capable of being an RDF consistency group, the composite group is added to PowerPath when a valid version of PowerPath is installed on the host and ONE of the following criteria is met:

- ◆ The `SYMAPI_RDF_CG_TO_PPATH` SYMAPI options file setting is set to ENABLED.
- ◆ The `-ppath` option parameter is supplied with the `symcg` command.
- ◆ The `symcg enable` command is called.

For information on enabling and disabling RDF consistency groups, refer to the *EMC Solutions Enabler Symmetrix SRDF Family Product Guide*.

When a composite group is created with a type of RDF1 or RDF2 and the composite group is put into PowerPath the type will be always be set to N/A until the first STD device is added to the group. A composite group with a type of N/A can only become an RDF1 or RDF2 composite group. Therefore, only R1 or R2 STD devices can be added to a composite group with type N/A.

Note: BCVs and VDEVs do not get stored in PowerPath.

Creating composite groups with multi session consistency (MSC)

When a composite group is created, it will be added to the RDF daemon for MSC when the following criteria is met:

- ◆ The `-rdf_consistency` option parameter is supplied at the time `symcg create` is called.
- ◆ All devices added must be in ASYNC mode when `symcg enable` command is called.

For information on enabling and disabling RDF consistency groups, refer to the *EMC Solutions Enabler Symmetrix SRDF Family Product Guide*.

Creating composite groups without consistency

When a composite group is created, it will be added to the SYMAPI configuration database only when one of the following criteria is met, regardless of whether PowerPath is installed on the host:

- ◆ The `SYMAPI_RDF_CG_TO_PPATH` options file setting is DISABLED (this is the default behavior and requires no configuration).
- ◆ The `-apidb` parameter is supplied with the `symcg` command.

When a composite group is created without consistency, a flag is set indicating so.

Moving a composite group to or from MSC or PowerPath

If a user wants to move a composite group from or to MSC or PowerPath they will need to export the composite group to a file, delete the composite group and then import the composite group with `-apidb` or `-ppath`.

Exporting a composite group to file

Use the following syntax to export a composite group to file:

```
symcgc -f Filename [-rdf] export CgName
```

The exported file will be formatted as follows:

```
<cg_type>
<dev_type> <symid_local_to_cg> [<rdfg> <remote_rdfg>] <symmetrix dsyno>
```

Valid values for `cg_type` include:

```
CGTYPE_REGULAR
CGTYPE_RDF1
CGTYPE_RDF2
CGTYPE_NA
```

Valid values for `dev_type` include:

```
S = STD device
B = BCV device
G = Gatekeeper
V = VDEV device
W = RVDEV device
R = RBCV device
Z = BRBCV device
Y = RRBCV device
T = TGT device
X = RTGT device
```

If `-rdf` is specified, the remote partners of the STD devices and BCV devices will be added to the file instead of those devices that exist in the current local composite group. The RBCV devices will become local BCVs. Non-RDF BCVs, VDEVs, BRBCVs, and RRBCVs will be ignored in this case.

Note: The `-rdf` option cannot be used on REGULAR composite groups.

Deleting a composite group

The following syntax is used to delete a composite group:

```
symcg delete CgName [-force] [-symforce]
```

This command deletes an existing composite group. If the composite group has members, the command will fail, unless the `-force` flag is used. If the force flag is specified, the devices that are members of the group are removed, and the group is deleted.

If PowerPath RDF consistency is enabled and cannot be disabled, the command will fail unless the `-symforce` flag is used.

If the CG is enabled for MSC, you cannot delete it unless the `-force` option is specified. Specifying `-force` does not disable the CG. You must use `symcg disable` to disable the CG.

Importing a Composite Group

The following syntax is used to import a composite group from a previously generated file:

```
symcg import CgName -f Filename  
[-apidb | -ppath | -rdf_consistency] [-rename]
```

If the `-apidb` parameter is specified then the composite group will be created in the SYMAPI configuration database only. If the `-ppath` parameter is specified then the composite group will be created in the SYMAPI configuration database and added to PowerPath. If the `-rdf_consistency` parameter is specified then the composite group will be created in the SYMAPI configuration database and added to RDF daemon.

Use the `-rename` option to have the operation result in a newly named composite group.

Adding standard devices

Once a composite group has been created, standard (STD) devices can be added to it.

Adding standard devices

The following syntax can be used to add a STD device to a composite group:

```
symcg add pd PdevName [LdevName] -cg CgName
```

or

```
symcg add dev SymDevName [LdevName] [-vdev | -tgt] [-rdf [-rdfg GrpNum]]  
-cg CgName -sid SymmID
```

When adding devices by specifying a Symmetrix device name, the Symmetrix array's ID (`-sid`) is required and additional options provide greater controls.

Composite groups will allow remote BCV devices (RBCVs) on both links to be associated with the RDF group. These RBCVs may be R1 or R2 type BCVs and may also have a remote BCV (RRBCV) associated behind each.

When the `-vdev` option is specified, the devices are added to the virtual device list. For this action to succeed, the device must already be configured as a VDEV. In addition, to add a VDEV to the remote virtual device list, specify the `-vdev -rdf` options.

When the `-tgt` option is specified, the devices are added to the target device list. In addition, to add a target device to the remote target device list, specify the `-tgt -rdf` options.

RDF group selection option, `-rdfg GrpNum`, can be used on non-concurrent RDF or concurrent RDF devices within a composite group to limit the scope of the action. A composite group could have two or more non-concurrent RDF devices on the same local Symmetrix but in two or more RDF groups. The user could restrict the operation to only those devices in the specified RA groups. Similarly, a composite group could have concurrent RDF

devices, which would each, by definition, be in two RDF groups. The user could use RDF group selection to restrict the operation to only one RDF group of the concurrent RDF device.

RDF group selections always refer to the locally attached Symmetrix arrays and RA group even if they are controlling devices or links that are not on the locally attached Symmetrix. However, when the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa.

Adding all devices

The following syntax can be used to add all devices that belong to a specified Symmetrix array and are not already members of a composite group to a specified composite group.

```
symcg addall -cg CgName [-sid SymmID] [pd | dev [-rdf]] [-R1|-R2]
[-RANGE <start:end>] [-RDFG GrpNum] [-vdev|-tgt]
```

If `-vdev` is specified, all VDEVs for the specified Symmetrix will be added to the virtual device list of the composite group. If `-tgt` is specified all devices will be added to the Clone target list of the composite group.

Note: You cannot add devices to a CG that has been enabled for RDF MSC.

Removing standard devices

Devices may be removed from an existing composite group. Device can either be removed individually, all devices, or all devices meeting the specified type can be removed.

Removing standard devices

The following syntax is used to remove a STD device from a composite group:

```
symcg remove dev SymDevName -cg CgName -sid SymmID [LdevName] [-force]
[-symforce] [-rdf] [-rdfg GrpNum]
```

or

```
symcg remove [pd PdevName | ld LdevName] -cg CgName [-force] [-symforce]
```

or

```
symcg remove -cg CgName [-force] [-symforce]
```

When removing devices by specifying a Symmetrix device name, the Symmetrix array's ID (`-sid`) is required. RDF group selection option, `-rdfg GrpNum`, can be used on non-concurrent RDF or concurrent RDF devices within a composite group to limit the scope of the action. A composite group could have two or more non-concurrent RDF devices on the same local Symmetrix but in two or more RDF groups. The user could restrict the operation to only those devices in the specified RA groups. Similarly, a composite group could have concurrent RDF devices, which would each, by definition, be in two RDF groups. The user could use RDF group selection to restrict the operation to only one RDF group of the concurrent RDF device.

RDF group selections always refer to the locally attached Symmetrix arrays and RA group even if they are controlling devices or links that are not on the locally attached Symmetrix. However, when the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa.

If PowerPath RDF consistency is enabled on a device you wish to remove, the command will fail unless the `-force` flag is used. If RDF consistency cannot be disabled, the command will fail unless the `-symforce` flag is used.

Removing an MSC RDF consistency-enabled device does not disable the device. You must use `symcg disable` to disable the CG.

Removing all standard devices

The following syntax is used to remove all STD devices from a composite group:

```
symcg rmall -cg CgName [-sid SymmID] [-RANGE <start:end>] [-force]
[-symforce] [-RDFG GrpNum] [-vdev|-tgt] [-rdf]
```

If `-vdev` is specified, only all VDEVs will be removed from the VDEV device list. If `-tgt` is specified, only all devices in the target device list will be removed.

If RDF consistency is enabled on a device you wish to remove, the command will fail unless the `-force` flag is used. If RDF consistency cannot be disabled, the command will fail unless the `-symforce` flag is used.

When the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa.

Renaming a composite group

To rename a composite group, enter:

```
symcg rename OldCGName NewCGName
```

Note: You cannot rename any PowerPath CG. Also, you cannot rename a MSC RDF consistency CG that has been enabled.

Setting RDF group attributes

Set a logical name to be associated with RDF groups. If the `-rdfg` argument is not specified then the action is applied to all RDF groups. Use the following syntax to set the naming:

```
symcg -cg CgName
set <[-name name] [-recovery_rdfg rag]>
[-rdfg <symid:<rag[,rag,...]|all>[,...] | name:name[,name]>]
```

This command provides a mechanism to group multiple RA groups within a specified CG to perform operations on all associated devices by specifying the RDF group name.

Composite group creation and output example

The `symcg` command can be used to obtain information about the composite groups visible to your host system. Use the following syntax to list the host-visible composite groups

The `-inactive` option is available for use in GNS-enabled environments to return the list of composite groups from the inactive group definition list. For information on active and inactive group lists in GNS, refer to [“Active vs. inactive group lists” on page 7-6](#).

The following example creates a non-consistent RDF1 composite group:

```
symcg create MyNewCG -type rdf1
```

RDF1 devices are added to our `MyNewCG` example:

```
symcg -cg MyNewCG addall dev -range CEE:CFD -sid 79  
symcg -cg MyNewCG addall dev -range 2:7 -sid 32
```

Then, RBCV devices are associated with the `MyNewCG` composite group:

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

In addition, RRBCV devices are associated with the `MyNewCG` composite group:

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

As a result, the example `list` command returns the following output:

symcg list

C O M P O S I T E G R O U P S								
Name	Type	Valid	Number of		Number of			TGTs
			Symms	RAGs	Devs	BCVs	VDEVs	
MyNewCG	RDF1	Yes	1	2	32	0	0	0

Use the `show` action to return additional details about the `MyNewCG` composite group. As a result, the example `list` command returns the following output:

Composite group lists

Use the following syntax to list the available composite groups:

```
symcg [-offline] [-v] list [-apidb | -ppath | -rdf_consistency]
```

If `-apidb` is specified only those composite groups that are not in PowerPath will be listed. With the `-ppath` option, lists those composite groups that exist in PowerPath are listed. When the `-rdf_consistency` parameter is specified, only those devices that exist in the RDF daemon are listed. The following is an example list operation:

symcg list

```

      C O M P O S I T E      G R O U P S

Name                Type    Valid   Number of
                        Symms   RAGs    Devs    Number of
                        Symms   RAGs    Devs    BCVs    VDEVs    TGTs
symcgtest2          REGULAR Yes      1      1      1      2      0      2
test_cg             RDF1    Yes      1      2      1      0      0      4
r1                  RDF1    Yes      1      1      0      1      0      1
test_lock           RDF1    Yes      1      1      0      0      2      2

```

Use the `show` action to return additional details about the a composite group. For example, to show the details of composite group `clonetgtcg`, enter:

symcg show clonetgtcg

```

Composite Group Name:  clonetgtcg
Composite Group Type   :  RDF1
Valid                  :  Yes
CG in PowerPath        :  No
CG in GNS              :  Yes
RDF Consistency Protection Allowed :  No
RDF Consistency Mode   :  NONE
Concurrent RDF         :  No
Number of RDF (RA) Groups :  1
Number of STD Devices   :  1
Number of CRDF STD Devices :  0
Number of BCV's (Locally-associated) :  0
Number of TGT's (Locally-associated) :  1
Number of VDEV's (Locally-associated) :  0
Number of RVDEV's (Remotely-associated VDEV) :  0
Number of RBCV's (Remotely-associated STD-RDF) :  0
Number of BRBCV's (Remotely-associated BCV-RDF) :  0
Number of RRBCV's (Remotely-associated RBCV) :  0
Number of RTGT's (Remotely-associated) :  1
Number of Symmetrix Units (1):
{
1) Symmetrix ID          :  000187940256
   Microcode Version     :  5671
   Number of STD Devices   :  1
   Number of CRDF STD Devices :  0
   Number of BCV's (Locally-associated) :  0
   Number of TGT's (Locally-associated) :  1
   Number of VDEV's (Locally-associated) :  0
   Number of RVDEV's (Remotely-associated VDEV) :  0
   Number of RBCV's (Remotely-associated STD-RDF) :  0
   Number of BRBCV's (Remotely-associated BCV-RDF) :  0
   Number of RRBCV's (Remotely-associated RBCV) :  0
   Number of RTGT's (Remotely-associated TGT-RDF) :  1

Number of RDF (RA) Groups (1):
{
1) RDF (RA) Group Number :  9          (08)

```

```

Remote Symmetrix ID      : N/A
Microcode Version       : N/A
Recovery RA Group       : N/A          (N/A)
RA Group Name           : N/A
STD Devices (1):
{

```

LdevName	PdevName	Sym Dev	Device Config	Sts	Flags CSR	Cap (MB)
DEV001	N/A	0314	RDF1+Mir	NR	.--	1031

```

}
Remote Target Devices (1):
{

```

LdevName	PdevName	Sym Dev	Device Config	Sts	Flags CSR	Cap (MB)
RTGT001	N/A	0212	STD	RW	.--	1031

```

}
Target Devices (Locally-associated non-RDF) (1):
{

```

LdevName	PdevName	Sym Dev	Device Config	Sts	Flags CSR	Cap (MB)
TGT001	/dev/rdisk/c6t0d76s2	0248	STD	RW	---	1031

```

}
Legend:

```

RDFA Flags:

C(onsistency) : X = Enabled, . = Disabled, - = N/A

(RDFA) S(tatus) : A = Active, I = Inactive, - = N/A

R(DFA Mode) : S = Single-session mode, M = MSC mode, - = N/A

Changing the state of devices in a composite group

The following commands provide the ability to change a devices state to write-disable, read/write enable, ready, not-ready, or relabel a device state. Use the following syntax to change the state of the devices in the a specific composite group:

```
symcg -cg CgName [-noprompt] [-v] [-force] [-bcv | -vdev | -tgt]
      [-star] [-sid SymmID]

write_disable [LdevName [LdevName...]]

rw_enable [LdevName [LdevName...]]

relabel [LdevName [LdevName...]]

symcg -cg CgName [-noprompt] [-v] [-force]
      [-bcv | -rbcv | -brbcv | -rrbcv | -vdev | -rvdev | -tgt | -rtgt]
      [-star] [-sid <SymmID> |
-rdfg <symid:<GrpNum[,GrpNum,...]|all>[,...] | name:name[,name,...]>]

ready [LdevName [LdevName...]]

not_ready [LdevName [LdevName...]]

hold [LdevName [LdevName...]]

unhold [LdevName [LdevName...]]
```

By default, this command, with no options specified, will write-disable all devices in the STD device list. Use the options provided to limit the action to a specific set of devices. Specify one or a list of logical device names (*LdevName*) to perform the action on a limited number of devices within the CG.

SYMAPI optional behavior parameters for composite groups

The SYMAPI_RDF_CG_TO_PPATH SYMAPI options file setting will be used to indicate whether or not the RDF composite groups should be added to PowerPath or not. By default, this option is disabled, and must be enabled to automatically add RDF composite groups to PowerPath.

Note: Options specified on the command line take precedence over settings configured in the options file.

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

This chapter describes the optional Group Name Services (GNS) feature.

◆ Group name services overview	7-2
◆ GNS configuration.....	7-7
◆ GNS daemon control.....	7-9
◆ How to mirror remote device group definitions.....	7-13
◆ Identify GNS groups	7-16
◆ GNS backup and restore mechanism.....	7-17
◆ Troubleshooting GNS.....	7-18

Group name services overview

Group name services (GNS) provides a common repository to store and maintain SYMAPI device group (DG) and composite group (CG) definitions across Symmetrix arrays that are visible to all locally attached hosts. By default, with GNS disabled, group definitions are stored in the local SYMAPI configuration database file on the host that created the group. Enabling GNS enables group definitions to be stored on the Symmetrix array in a shared GNS repository. This shared GNS repository is visible to any GNS-enabled locally-attached host enabling these hosts to perform control operations, regardless of which host initially defined the group. In addition, if one host goes down, you can still perform SYMCLI control operation from another local host in your Symmetrix environment.

How the GNS daemon works

In the GNS-enabled environment, each host performing management operations must run an instance of the GNS daemon (`storgnsd`). In this case, the Solutions Enabler SYMAPI and SYMCLI do not directly access the GNS shared repository. Instead, requests are forwarded to the GNS daemon, which processes all GNS operations. This daemon is the only entity that directly accesses the GNS shared repository and is responsible for ensuring that each host has access to the most current GNS definitions.

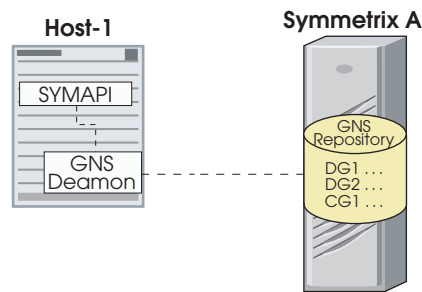


Figure 7-1 GNS daemon controls GNS activity

From each host, a GNS daemon listens for GNS requests from local clients (same host) and carries them out on the locally attached Symmetrix array. In addition, the GNS daemon monitors the GNS repositories on all locally-attached Symmetrix arrays, at a user-configured polling interval, for changes made to the shared GNS repository by other daemons (on other hosts) as show in [Figure 7-1](#). When a change is identified, the GNS daemon will update the host to ensure that all GNS-enabled hosts refer to the same group definitions.

A set of options are available for controlling the GNS daemon. For information on configuring the GNS daemon refer to [“GNS daemon options file” on page 7-10](#).

Shared group definitions with GNS

GNS provides a common repository to share device group and composite group definitions across hosts locally-attached to the same set of Symmetrix arrays. A *device group* is a user-defined object comprised of devices that belong to a single Symmetrix array and a single RA group. In a GNS-enabled environment, device group definitions are stored in the GNS repository of the Symmetrix array on which the devices within the device group reside and are visible to all hosts locally attached to that Symmetrix array.

A *composite group* is also a user-defined object comprised of devices; however, the device members of a composite group can belong to multiple Symmetrix arrays and RA groups. In a GNS-enabled environment, composite group definitions are distributed across all Symmetrix

arrays that contain device members of the composite group by the GNS daemon. A host must be locally attached to all arrays containing devices in the composite group to manage or control that composite group. If a host is only attached to a subset of the Symmetrix arrays that a composite group spans, that group will be visible to the host but in an invalid and unusable state. This is not a recommended configuration.

The GNS state that is visible from a given host is determined by the set of arrays to which the host is connected. As seen in [Figure 7-2](#), Host-1 is GNS-enabled and locally-attached to Symmetrix A, Symmetrix B, and Symmetrix C. Host-1 can access and modify the group definitions on all three arrays. The group definitions that are visible to Host-1 include device groups DG1, DG2, DG3, DG4, DG5, and DG6 and composite group CG1's definition, which contains devices on Symmetrix B and C.

Host-2 is GNS-enabled and locally-attached to Symmetrix B and Symmetrix C. The group definitions that are visible to Host-2 include device groups DG3, DG4, DG5, and DG6 and composite group CG1. Since Host-2 is locally attached to only Symmetrix array B and C, it cannot see any device groups or composite groups on Symmetrix A (that is, DG1 and DG2).

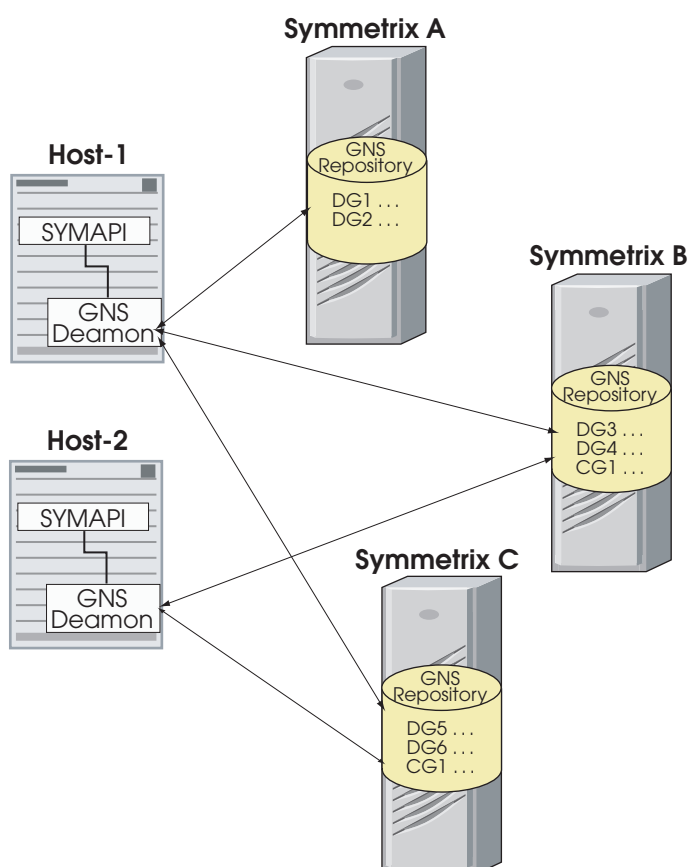


Figure 7-2 Host-visible GNS state

How GNS handles updates to group definitions

When GNS is enabled, device group and composite group definitions are stored in the common GNS repository.

For example, in [Figure 7-3](#), Host-1 makes a change to composite group CG1. (1) The Host-1 GNS daemon gets this request to process and (2) updates the GNS shared repository on all relevant Symmetrix arrays, in this case Symmetrix A, B, and C. The GNS daemon running on Host-3 detects the change (3) while polling its locally-attached GNS repositories (attached Symmetrix A, B, and C) and updates Host-3 (and PowerPath if necessary, refer to [“GNS and consistency groups” on page 7-5](#)) with the updates made by Host-1. Host-3’s client application will detect the change (4) on its next group call.

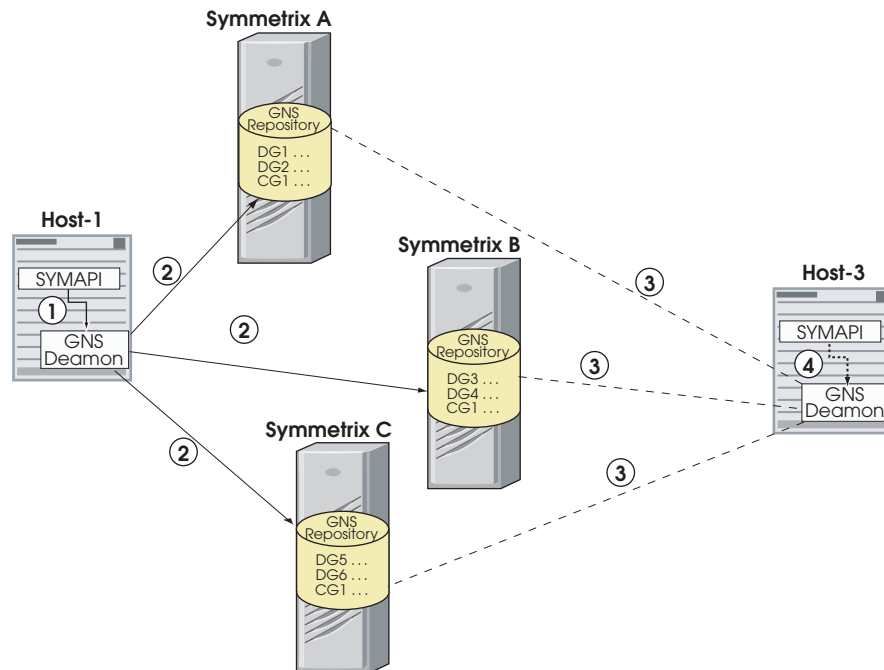


Figure 7-3 GNS across multiple hosts

GNS and consistency groups

Composite groups can be defined to work with PowerPath or Multi Session Consistency (MSC, which uses the RDF Daemon). In a GNS-enabled environment, changes made to any composite group marked for consistency are automatically propagated to PowerPath or the RDF Daemon on all relevant hosts. When updates are made to the GNS repository, the GNS daemon will update the local PowerPath database or RDF Daemon.

For example, if a device is added to a composite group that has been marked for PowerPath, the newly added device will also be added to each locally-attached host's PowerPath database by the GNS daemon on that host, assuming each host is running GNS. Similarly, if a device is removed from the composite group, it will be automatically removed from all relevant host's PowerPath databases. For more information on marking composite groups for PowerPath, refer to [“Creating composite groups with PowerPath” on page 6-19](#). For more information on marking composite groups for MSC, refer to [“Creating composite groups with multi session consistency \(MSC\)” on page 6-19](#).

Note: When using the `symcg create` command, the `-apidb` option can be used to override this behavior, allowing changes to only get committed to the SYMAPI configuration database.

GNS device groups and SRDF

In an RDF scenario, a local GNS-backed device group (either RDF1 or RDF2) can be automatically mirrored on the remote Symmetrix array via SRDF links — for activation and use during a disaster fail-over situation. By default, the GNS device group definitions are stored on the local (directly-attached) Symmetrix — where the local devices are located.

Note: This functionality is not supported for composite groups.

Optionally, any changes made to the local group definition (for example, adding a standard or BCV device) can be set to automatically maintain a mirrored group definition on the remote Symmetrix. This remotely mirrored group is for disaster recovery situations, where entire applications (including group aware ones), fail-over and are re-started on the remote side (where the remote devices are).

As changes are made to the local group definition, GNS automatically propagates corresponding changes to the remote array, so that the two are kept synchronized. For more information on configuring this option, refer to [“How to mirror remote device group definitions” on page 7-13](#).

Note: This functionality is only supported for SRDF device groups that do not have concurrent RDF devices.

GNS behavior in client/server mode

In client/server mode, client group operations are sent to the server, which may hand off these operations to GNS for servicing. In this case, the SYMAPI server acts as the GNS client, and the server's SYMAPI options file (SYMAPI_USE_GNS) determines whether GNS is enabled for the client. If GNS is enabled and the client is caching (SYMAPI_CONN_TYPE_REMOTE_CACHED), the client checks periodically with the server to determine if the group definitions that it is caching need to be updated.

The SYMAPI_GNS_CS_STALE_DATA_TIMEOUT option in the client's SYMAPI options file controls how frequently the client checks the server. The default is one second. When a client needs group information, it validates its cached group definitions with the server if the time since the last check is longer than the default value or whatever new value you have set for this option.

Active vs. inactive group lists

When GNS is enabled, group definitions are stored in the global GNS repository instead of within individual host-based configuration database files. In order to facilitate the importing of groups into GNS, a limited access to the group definitions held within a host-based configuration database is provided while GNS is enabled.

The groups within the individual host-based configuration database files can be listed via the following commands:

```
symdg list -inactive  
symcg list -inactive
```

A group contained in the host-based configuration database can be imported into GNS via the following commands:

```
symdg activate DgName  
symcg activate CgName
```

All groups in the host-based configuration database can be imported into GNS via the following commands:

```
symdg activateall  
symcg activateall
```

Note: Importing groups into GNS may fail if the group name is already in use within GNS.

Although the above commands can be used while GNS is disabled, their use is discouraged since in this situation it provides access to a copy of the GNS based group definitions written, for diagnostic purposes, into the host-based configuration database when GNS was enabled in the past.

GNS configuration

To take advantage of the GNS features, you must first configure your Symmetrix environment to support GNS.

How to set up the GNS option

GNS use is optional and can be either enabled or disabled on each host via the SYMAPI options file setting `SYMAPI_USE_GNS`. This option is enabled as follows:

```
SYMAPI_USE_GNS=ENABLE
```

The default setting is `DISABLE`. Setting this option to `ENABLE` enables GNS for any applications using the default SYMAPI configuration database file. It is assumed that applications running with a private configuration database file intend to keep their modifications private, and therefore, do not make use of GNS.

How to configure GNS in a multi-host environment

When configuring GNS in a multi-host environment, it is recommended that you first review the groups currently stored on all hosts that share access to a set of Symmetrix arrays. Resolve any potential conflicts with other hosts. Also, if using composite groups ensure that all hosts on which you intend to enable GNS are local to the same set of arrays to avoid composite groups appearing inconsistent to hosts that aren't attached to all referenced arrays.

When ready, enable GNS on a single array being sure to configure the any options or user authentication that you desire. For details on managing GNS use via the options file, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*. In addition to enabling GNS on the host, you can configure: GNS User Authentication (setup a specific set of users with access to the GNS daemon) and GNS Daemon Control (start and stop the daemon, and configure GNS daemon options).

Once GNS is enabled on a host, activate those groups that you wish to expose via GNS. Enable GNS on a second host. Validate that the new host can see the previously activated groups and resolve any conflicts with other hosts before activating additional groups.

GNS user authentication

GNS is intended to share group definitions across multiple users and hosts in a Symmetrix environment via the host installed GNS daemons. As such, access to GNS functionality is controlled by limiting permission to the GNS daemon. This access is controlled via the common daemon authorization file, `daemon_users`. This file is located in the following directories:

UNIX	<code>/var/symapi/config/daemon_users</code>
Windows	<code>c:\Program Files\EMC\SYMAPI\config\daemon_users</code>

Note: It is important to protect this file so that only privileged administrators can modify it.

Users meeting any of the following criteria will be permitted to control and use the GNS daemon:

- ◆ User with privileges; UNIX users with root access and Windows users that are a members of the Administrators group
- ◆ Users listed in the `daemon_users` file located on each host from which they require access

For example, the following lines in the `daemon_users` file would permit users named `Smith` and `Jones` to use the GNS daemon:

```
smith    storgnsd
jones    storgnsd
```

How to configure GNS for consistency

SRDF consistency groups (composite groups enabled for consistency) work in conjunction with PowerPath or MSC. If you define SRDF consistency groups in a GNS-enabled environment, the GNS daemon running on all relevant hosts will automatically update its local PowerPath device database or RDF Daemon for MSC as groups are changed. This will ensure that device definitions remain consistent.

GNS daemon control

There are three ways that the GNS daemon can be started. If GNS is enabled, the daemon will be started automatically by the Solutions Enabler libraries the first time they attempt to connect with it, which can cause a slight delay in performance on that initial connection while the daemon starts and builds its cache.

Note: Prior to starting `storgnsd`, ensure that your default SYMAPI configuration database is up-to-date, since `storgnsd` uses the information stored in it to establish contact with your Symmetrix arrays. Refer to [“Discovering configuration and status data” on page 2-2](#) for details.

Alternatively, the daemon can be started manually via the `stordaeomon` command line utility as follows:

```
stordaeomon start storgnsd
```

In addition, the daemon can be set to start automatically every time the local host is booted using the following command line:

```
stordaeomon install storgnsd -autostart
```

Pre-starting the daemon, either manually or via the automatic option, is useful because the daemon may take a while to initially construct its cache — depending on the number of groups and Symmetrix arrays it has to load.

If the daemon is stopped for some reason, it can optionally be restarted automatically by an internal Solutions Enabler watchdog mechanism. A combination of the watchdog mechanism and the auto-start option described above can be used to ensure that the daemon is always running, which is important with RDF consistency composite groups to ensure that PowerPath or the MSC RDF Daemon is updated whenever group changes are made.

Enabling the watchdog

This watchdog mechanism is enabled by default. The watchdog daemon will restart the GNS daemon if it crashes or is killed. This behavior can be disabled via the `storgns:autorestart` entry in the `daemon_options` file (for more information, refer to [“GNS daemon options file” on page 7-10](#)). On UNIX, this restart functionality is provided by a dedicated watchdog daemon (`storwatchd`). This is started automatically as needed. On Windows, this restart functionality is provided by the Service Control Manager.

Note: A maximum of three restarts within a 15 minute period will be attempted. Following a third restart (again, within a given 15 minute period), no subsequent restart will be attempted.

Managing the GNS daemon

The GNS daemon can be controlled by the existing `stordaeomon` utility. These control options include:

- ◆ Stopping the daemon.
- ◆ Querying the daemon's status.
- ◆ Querying the daemon's log files.

For details, refer to the `stordaeomon` man page.

GNS Daemon and Symmetrix external locks

The GNS daemon uses two Symmetrix External Locks (SELs) to maintain exclusive access to the GNS repository on each Symmetrix array: F0 and F1.

If the daemon encounters a needed SEL lock that has been held by someone for more than 1 minute, it assumes the lock is stranded and forcibly releases it.

The following commands can be used to view available GNS locks:

```
symcfg -sid nnnn -lockn GNS list
```

The following command can be used to manually release a GNS lock:

```
symcfg -sid nnnn -lockn GNS release
```

GNS daemon log file

The GNS daemon writes its log (trace) messages to the standard location used by all daemons. The locations are:

```
UNIX      /var/symapi/log/storgnsd.log0
           storgnsd.log1
```

```
Windows  c:\Program Files\EMC\SYMAPI\log\storgnsd.log0
           storgnsd.log1
```

These two files are written in an alternating, round-robin manner. When the active one becomes full, it is closed and the other one is truncated and made active.

The contents of the log file can also be displayed as follows:

```
stordaeomon showlog storgnsd -lines 200
```

GNS daemon options file

Configuration options for the GNS daemon are contained within the daemon options file (storgnsd) located in the following directories:

```
UNIX      /var/symapi/config/daemon_options
Windows  c:\Program Files\EMC\SYMAPI\config\daemon_options
```

Options can be specified in this file using the following syntax:

```
storgnsd:OptName = OptValue
```

Where the *OptName* is the name of the option and the *OptValue* is the new value you wish to set.

Editing or removing parameters manually

The *storgnsd* file contains a set of parameters that can be modified to affect GNS behavior via SYMCLI or SYMAPI commands. The file contains editable behavior parameters set to certain optional defaults in the line entries. Commented lines beginning with a pound sign (#) are ignored.

To remove any parameter option, remove the line entry, rename the file, or comment the line by adding a pound sign (#) at the beginning of the line entry.

Editing or removing parameters via the CLI

Option file values also can be set using the `stord daemon setoption` CLI utility, which programmatically makes changes to the `daemon_options` file. The following is the command-line syntax for this option:

```
stord daemon setoption DaemonName
-name OptName=OptValue [-force]
```

Where the *OptName* is the name of the option and the *OptValue* is the new value you wish to set.

The following example shows how this utility can be used to set or change an option setting:

```
stord daemon setoption storgnsd -name autorestart=enable
```

The following example shows how this utility can be used to remove an option setting:

```
stord daemon setoption storgnsd -name autorestart=
```

Note: In the above example an empty value has been passed.

By default, only options already present in the file (such as those in use or commented out) can be set. The `-force` option can be supplied to force a change, even if it requires adding a new option line to the file.

GNS daemon option file parameters

The following are possible optional parameter entries for the options file:

Table 7-1 GNS daemon optional behavior parameters

Optional behavior parameter	= <OptValue DefaultValue>	Description
storgnsd:autorestart	= disable enable	If set to <code>disable</code> , the watchdog mechanism is turned off and the daemon will no longer automatically restart if it crashes.
storgnsd:gns_device_poll_interval	= <i>nnn</i> 15	The interval, in seconds, at which to poll locally-attached Symmetrix arrays for group changes.
storgnsd:gns_device_remote_poll_interval	= <i>nnn</i> 60	The interval, in seconds, at which to poll remotely-attached Symmetrix arrays for group changes.
storgnsd:gns_init_remote_symms	= disable enable	By default (enable), GNS will initialize the GNS group repository on remote Symmetrix arrays while remotely mirroring groups (when the <code>gns_remote_mirror</code> option is enabled). If disabled, a GNS Daemon directly connected to such remote arrays has to perform the initialization when it encounters the array. This option should be disabled if all of the following criteria are met: • Remote mirroring is enabled (<code>gns_remote_mirror</code>) • GNS Daemons are actively running on both sides of the SRDF link • Groups are being modified on both the local and remote Symmetrix arrays • GNS backup/restore is enabled (<code>gns_shadow_file</code>). Failing to disable this option in this scenario may result in data loss due to interaction between the backup/restore and remote mirroring mechanisms.
storgnsdLogfile_size	= <i>nnn</i> 1000	The maximum size in KB that the logfile(s) are allowed to grow before they begin to overwrite themselves.
storgnsd:gns_ppath_poll_interval	= <i>nnn</i> 60	The interval, in seconds, at which to check whether PowerPath definitions need to be updated.
storgnsd:gns_remote_mirror	= enable disable	If set to <code>enable</code> , the daemon will attempt to mirror RDF device group definitions on remote Symmetrix arrays. For more information, refer to “How to mirror remote device group definitions” on page 7-13 .

Table 7-1 GNS daemon optional behavior parameters (continued)

Optional behavior parameter	= <OptValue DefaultValue>	Description
storgnsd:gns_rmtarr_update_interval	nn 60	The interval, in seconds, at which to check for device group (DG) changes that need to be propagated to remote Symmetrix arrays. Refer to “How to mirror remote device group definitions” on page 7-13 .
storgnsd:gns_shadow_file	= enable disable needed	If set to <code>enable</code> , the daemon automatically maintains backup copies of GNS data for locally attached Symmetrix arrays. These backups, maintained on the local disk, are automatically restored to the GNS repository on an array if the daemon detects that a repository has been erased (for example, in the event of a power failure under Enginuity 5x67). If set to <code>needed</code> , backups are only maintained for Symmetrix arrays with Enginuity versions lower than 5x71. Refer to “GNS backup and restore mechanism” on page 7-17 .
storgnsd:gns_symavoid	= symid,symid,...	A comma separated list of 12-digit Symmetrix IDs for arrays that the daemon should ignore. If this attribute is defined, the daemon will neither look for nor place GNS data on these arrays.
storgnsd:gns_symdiscover_interval	= nnn 600	The interval, in seconds, at which to check for Symmetrix configuration changes.
storgnsd:gns_syminclude	= symid,symid,...	A comma separated list of 12-digit Symmetrix IDs for arrays that the daemon should manage. If this attribute is defined, <i>only</i> these arrays — and no others — are used for GNS.

How to mirror remote device group definitions

The option `gns_remote_mirror` in the GNS daemon's options file determines whether GNS should attempt to remotely mirror a device group or a composite group RDF definition contained in the shared GNS repository for a given Symmetrix array.

When enabled, GNS will maintain a remote mirrored group definition with the same name as the local one creating a usable group to hosts (including GNS daemons) directly connected to the remote Symmetrix array(s). The remote mirrored group has the same name as the local one, and has a mirror image of its contents; in other words, the data reflects the perspective of the local array. This done to ensure that the mirrored group is a legal, usable group to hosts directly connected to the remote Symmetrix arrays.

Controlling the mirrored group

An RDF group that has been created by this mirroring mechanism is flagged internally by GNS as being a mirror.

By default, these mirrored groups are read-only and cannot be modified or renamed — although GNS will continue to update them as the local groups, upon which they are based, are changed. This behavior can be changed by setting the `SYMAPI_GNS_MIRRORED_GROUP_CONTROL` option in the SYMAPI options file to `ENABLE` (the default value is `DISABLE`).

If a mirrored group is directly modified (from a host connected to the remote Symmetrix on which it is defined), the connection between it and the local group, on which it was based, is broken. At that point it is no longer a mirror and changes to its base group (the local group) are no longer propagated to it. If a mirrored group is renamed or deleted, GNS will subsequently recreate the mirrored group from its base group.

Impact of remotely mirroring group definitions

The following changes are made to the remote group's definition:

Note: For the RDF remote mirroring of CGs, the BCV devices (both local and remote; BCV, RBCV, BRBCV, RRBCV) are not mirrored and therefore will not appear in the group created on the remote host.

- ◆ The remote group definition maintains the same name and type as the local group.
- ◆ The group sub-type is reversed: an RDF1 local group becomes an RDF2 remote one; an RDF2 local group becomes an RDF1 remote one.
- ◆ Attributes that are unrelated to particular devices or storage arrays are copied to the remote copy without change. For example, modification time, vendor, and group name.
- ◆ The remote RA group number (RRAGRP) attribute is changed on the remote side to contain the corresponding local RDF group number.
- ◆ A number of attributes come in pairs: local and remote. On the remote side, these are swapped. For example:
 - The local and remote standard (DEVS and RDEVs), the local and remote BCV device lists (BCVs and RBCVs), and remote BCVs of the remote BCV device lists (BRBCVs and RRBCVs) are swapped.

- ◆ The device logical device name list is left unchanged. The device and remote device lists are swapped as noted above, but the logical device name list is not; the same names apply to the local device in both the original and mirrored copy.

Note: This works because the size of the Local and Remote standard device lists are the same.

- ◆ The BCV and remote BCV logical device name lists are swapped and a simple name transformation is applied.

BCV and remote BCV device alias name swap

In the remote mirror name transformation, the BCV alias list becomes the RBCV alias list in the remote mirror. Alias names that have a value of `BCVnnn` are automatically changed to `RBCVnnn`.

In turn, the RBCV alias list becomes the BCV alias list in the remote mirror. In this case the alias names that have a default value of `RBCVnnn` are automatically changed to `BCVnnn`.

Similarly, the RRBCV logical list becomes the BRBCV list in the remote mirror. Logical names that have a default value of `RRBCVnnn` are automatically changed to `BRBCVnnn`.

As expected, the BRBCV logical list becomes the RRBCV list in the remote mirror. Logical names that have a default value of `BRBCVnnn` are automatically changed to `RRBCVnnn`.

These names are the default names (logical device names) supplied by SYMCLI if the user doesn't provide one. If the user has overridden the defaults and supplied their own names that look like the above, they will still be transformed as described.

In the unlikely scenario that the user has explicitly supplied aliases, for example, `BCVnnn` and `RBCVnnn`, it is possible that the above transformation may result in duplicate alias names in the remote mirror. For example, if the original BCV alias list contained `BCV001`, `RBCV001`, the remote mirror's RBCV alias list would contain `RBCV001`, `RBCV001`, which would cause a name conflict.

The following example illustrates the mirrored group data:

--- Local DG Group, on Symm1 ---		--- Remote DG Group, on Symm2 ---	
Type	RDF1	Type	RDF2
Role	Have a Remote Mirror	Role	Am a Remote Mirror
Remote Symm	Sym2	Remote Symm	Sym1
Local RDFGrp	2	Local raGrp	5
Remote RDFGrp	5	Remote raGrp	2
Devices	0001,0002,0003	Devices	0091,0092,0093
Remote-Devs	0091,0092,0093	Remote-Devs	0001,0002,0003
LDevNames	DEV001,DEV002,DEV003	Dev Aliases	DEV001,DEV002,DEV003
BCVs	0010,0011	BCVs	0020,0021,0022
Remote-BCVs	0020,0021,0022	Remote-BCVs	0010,0011
BCV LdevName	BCV001,BCV002	BCV Aliases	BCV001,BCV002,BCV003
RBCV LdevName	RBCV001,RBCV002,RBCV003	RBCV Aliases	RBCV001,RBCV002
Vendor	EMC	Vendor	EMC
ModTime	12345	ModTime	12345
GateKeepers	/dev/rdisk/c2t0d2		

Figure 7-4 Mirrored Remote Group Definition

Composite group logical device name support

A composite group can contain logical device names (aliases) for devices. However, CGs created with Solutions Enabler versions prior to 6.0 will not contain LdevNames and therefore you will not be able to use command options that contain LdevNames.

All new CGs, and any devices added to them, will automatically be assigned an LdevName if you do not specify one.

Note: Hosts running prior versions of Solutions Enabler have no knowledge of these device alias names and will not provide them when a consistency group is modified.

Identify GNS groups

Once you have configured your Symmetrix environment to support the GNS option. You can begin to create and modify groups. Using the standard `symdg show DgName`, for device groups, or `symcg show CgName`, for composite groups, to determine if a group is stored in the GNS group list.

Group output

For a device group, the output of the command `symdg show DgName` identifies whether a group definition is stored in GNS. The sample output below shows that group `MyRegDeviceGroup` is stored in the GNS list since the output parameter **Device Group in GNS** has a value of **Yes**. It also shows the device group's GNS mirror state.

```
% symdg show MyRegDeviceGroup

Group Name: MyRegDeviceGroup

    Group Type                : REGULAR
    → Device Group in GNS      : Yes (Is Mirror)
    Valid                      : Yes
    Symmetrix ID               : 000184500160
    Group Creation Time        : Tue Aug 4 16:44:18 2006
    Vendor ID                  : EMC Corp
    Application ID             : SYMCLI

    Number of STD Devices in Group : 20
    Number of Associated GK's       : 0
    Number of Locally-associated BCV's : 20
    Number of Locally-associated VDEV's : 0
    . . .
```

In addition, for a composite group, the output of the command `symcg show CgName` identifies whether a group definition is stored in GNS. The sample output below shows that group `MyCompGroup` is stored in the GNS list since the output parameter **CG in GNS** has a value of **Yes**. It also shows the composite group's GNS mirror state. The group definition has not been propagated to PowerPath since the output parameter **CG in PowerPath** has a value of **No**, but the group has been enabled for RDF consistency via MSC.

```
% symcg show MyCompGroup

Composite Group Name: MyCompGroup

    Composite Group Type      : RDF1
    Valid                    : Yes
    → CG in PowerPath         : No
    → CG in GNS               : Yes (Is Mirror)
    → RDF Consistency Protection Allowed : Yes
    → RDF Consistency Enabled   : Yes

    Number of RDF (RA) Groups : 4
    Number of STD Devices     : 64
    Number of BCV's (Locally-associated) : 0
    Number of VDEV's (Locally-associated) : 0
    Number of RBCV's (Remotely-associated STD-RDF) : 0
    Number of BRBCV's (Remotely-associated BCV-RDF) : 0
    Number of RRBCV's (Remotely-associated RBCV) : 0
    . . .
```

GNS backup and restore mechanism

GNS group definitions are stored within the each individual Symmetrix array. These definitions can be lost if a Symmetrix is completely re-initialized or power is lost on a Symmetrix running Enginuity 5x67.

A GNS daemon can be configured to automatically backup and restore GNS data from individual arrays. When backup is enabled, The daemon will write an image of the GNS group information to the host's local disk each time it makes a change or it notices a change made from another locally attached host. These backups are maintained as separate per-Symmetrix files, named *SID.shadow*, where *SID* is the Symmetrix array ID, and stored in the following directory on the host machine:

UNIX	<code>/var/symapi/gns/<i>SID</i>.shadow</code>
Win32	<code>c:\Program Files\EMC\SYMAPI\gns\<i>SID</i>.shadow</code>

Note: Because the maintenance of numerous backup images can impact performance, it is recommended that you do not enable it on all GNS daemons. Typically, it is recommended that you configure a maximum of 2 or 3 GNS daemons to maintain backups for each Symmetrix array that may hold GNS data.

Automatic GNS backups

An option in the GNS daemon's options file determines whether it will perform automatic backup and restore. To enable GNS backups, enter:

```
storgnsd:gns_shadow_file = always
```

The default is `disable`. For more information on the GNS options file, refer to [“GNS daemon options file” on page 7-10](#).

Manual backups and restores via storgnsd

Backups to and restores from the shadow files can be manually triggered by invoking the `storgnsd` program as follows.

To generate a GNS backup file for the specified Symmetrix array enter:

```
storgnsd backup -sid SID
```

To generate a GNS backup file for all locally attached Symmetrix arrays, enter:

```
storgnsd backup
```

To restore GNS data to the specified Symmetrix array from a backup file previously created, enter:

```
storgnsd restore -sid SID [-noprompt]
```

Since this operation is destructive in nature, a confirmation is required prior to completing the operation. Using the `-noprompt` argument will skip the confirmation step.

Troubleshooting GNS

When using GNS, you should be aware of the possibility of group name collision and inconsistent groups and what SYMAPI does to rectify each scenario.

Group name collisions

Each GNS daemon attempts to ensure that group names are unique (relative to a group type). It is possible, however, for duplicate named groups to occur.

For example, it is possible for two different groups to be created with the same name on different Symmetrix arrays. A subsequent consideration change could then allow the GNS daemon to see both groups.

If a GNS daemon detects duplicate group names, it returns a state flag informing clients of this fact and modifies the group names to be unique by attaching a numeric suffix. For example, duplicate device groups with the name `MyDG` could be modified in the following way:

```
MyDG#12345678
MyDG#87654321
```

While in this state (duplicate names), the only operations permitted on them are those that can be used to resolve the name conflict: group rename and delete (with the `-force` flag). For example:

```
symdg delete MyDG#12345678 -force
symdg rename MyDG#87654321 MyDG
```

Note: For information on renaming composite groups, see [“Renaming a composite group” on page 6-23](#).

If users at two different hosts simultaneously create groups of the same type with the same name using devices on different Symmetrix units, both attempts might succeed. Because each host's GNS daemon is polling for changes made elsewhere, the daemons might not detect the name conflict until after the fact.

Inconsistent groups

A composite group is placed into an inconsistent state if one or more of the Symmetrix arrays that it resides on (in a GNS-enabled environment) cannot be reached. For example, each Symmetrix array that a consistency group spans records the identity of the other arrays that the group spans. If a discrepancy is noticed where a composite group is not defined on an array, even though some other array indicates it should be there, that group is assumed to be inconsistent or invalid.

Groups are also placed into an inconsistent state if certain internal bookkeeping information maintained by GNS is found to be missing or incorrect. For example, a consistency group spans two arrays and the GNS state recorded on each Symmetrix verifies that the group also resides on the other array. If at some point one array is found to not contain the group (while the other array still thinks it should be found there), the group is assumed to be inconsistent, or invalid.

This could happen if one of the arrays is re-initialized or a host attached only to one array decides to delete the composite group there. Since the group definition is also stored on an unreachable array, that group state is seen as invalid, or inconsistent, from that host. A user there would have to use the `-force` flag to delete the group. Groups in the inconsistent state can be deleted with the `-force` flag.

This chapter describes how to use SYMCLI to retrieve the performance-related statistics such as the number of I/O requests and throughput activities for the Symmetrix devices.

◆ Overview.....	8-2
◆ REQUESTS statistics.....	8-4
◆ BACKEND statistics and PREFETCH activity	8-9
◆ Director statistics	8-10
◆ PORT statistics	8-12
◆ MEMIO statistics	8-13
◆ DMSP statistics.....	8-14
◆ Disk statistics.....	8-15
◆ CACHE statistics	8-16
◆ PATH statistics	8-20
◆ CYCLE statistics.....	8-22

Overview

The Symmetrix array maintains statistical counters for performance-critical operations. Using the `symstat` command, you can view the performance statistics derived from these counters.

The statistics command (`symstat`) performs the following:

- ◆ Queries Symmetrix devices to capture raw performance counts and store them in memory
- ◆ Retrieves the performance counts for the Symmetrix array as a whole
- ◆ Retrieves the performance counts for a director or director port
- ◆ Retrieves the performance counts for one or more Symmetrix devices
- ◆ Retrieves the performance counts for one or more Symmetrix device groups, composite groups, or RDF groups
- ◆ Retrieves the performance counts for a selection of, or all, Symmetrix disks
- ◆ Retrieves the time stamp of the performance count sample
- ◆ Retrieves and displays replication session statistics for SRDF/A

Note: Performance counters are not stored in the configuration database and only exist in the session context.

About the `symstat` command

The `symstat` command captures, in real time, statistics information about a Symmetrix array. You can examine the performance of one or more devices, a device group, or any or all directors. For a capture or sample period, `symstat` only displays devices or directors that had activity.

The `-c` argument defines the number of samples. The default for this argument is continuous sampling. If you do not specify this argument but you specify an `-i` value, the command produces a continuous statistical output, requiring a cancel (Ctrl-C) to stop the process. If no sample interval is specified, the default sample interval is 10 seconds; the minimum is 5 seconds. It is recommended that 30 seconds or greater be used for effective statistical sampling.

Reporting objects

The SYMCLI `symstat` command allows you to return valuable statistics on the how the following objects are performing in your environment:

- ◆ **Devices** — Statistical data can be returned on device throughput and I/O. Also, throughput activity to a specific set of device within a device group can be obtained in addition to cache memory to disk activity for selected devices. And dynamic mirroring service policy (DMSP) statistics for the selected device(s) can be obtained.
- ◆ **Directors** — Throughput, backend I/O, and port activity statistics can be obtained for a specific director or type of director.
- ◆ **Disk** — Reports back-end I/O requests and throughput for selected disks and track prefetch disk activity for selected back-end directors only.
- ◆ **SRDF/A sessions** — Monitor activity of SRDF/A sessions using the CYCLE, CACHE, and REQUESTS types in conjunction with the `reptype rdfa` flag of the `symstat` command.

For more details about performance types, refer to [“Performance type options” on page 8-3](#).

Performance type options

The `symstat` command provides different types of performance display options:

Table 8-1 Performance display options

Option	Description
REQUESTS	Reports I/O requests and throughput for selected devices, directors, or SRDF/A sessions. (This is the default type; if no type is specified REQUESTS is used.)
BACKEND	Reports back-end I/O requests and throughput for selected devices.
PORT	Reports performance statistics for a director port.
CACHE	Reports cache activity for selected front-end or remote link directors, or SRDF/A sessions.
MEMIO	Reports cache memory to disk activity for selected devices.
PATH	Report R-Copy path information for non-incremental sessions. Symmetrix arrays that have all or some incremental sessions will report an error.
CYCLE	Report cycle summary information for SRDF-A sessions.
DISK	Reports back-end I/O requests and throughput for selected disks.
PREFETCH	Reports track prefetch disk activity for selected back-end directors only.
DMSP	Reports dynamic mirroring service policy (DMSP) statistics for the selected device(s).

REQUESTS statistics

Using `symstat`, you can display performance statistics such as I/O requests and throughput for the devices within a Symmetrix array, a device group, or SRDF/A sessions by specifying the REQUESTS type. REQUESTS is the default type for `symstat` operations, therefore, if no type is specified, REQUESTS performance information is returned.

Array statistics

I/O and throughput statistics can be returned for all devices on a specific array by specifying the Symmetrix ID in the command-line, specifying the interval and the number of samples (counts) for the statistical run. For example, for sample intervals of 5 seconds and a sample count of 2 for I/O requests and throughput on Symmetrix 097. In this example, the performance type is not specified and, therefore, defaults to REQUESTS. Using this command, the following performance display is an example for a statistical run on a Symmetrix array:

```
symstat -i 5 -c 5 -sid 097
```

09:03:11		DEVICE		IO/sec		KB/sec		% Hits		%Seq	Num	WP
				READ	WRITE	READ	WRITE	RD	WRT	READ	Tracks	
1		2		3	4	5	6	7	7	7	7	
09:03:28		7D0F	(Not Visible)	0	0	0	0	N/A	N/A	N/A	4	
		7D3F	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D40	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D41	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D42	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D43	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D44	(Not Visible)	0	6	0	4	N/A	100	N/A	2	
		7D45	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D46	(Not Visible)	0	3	0	1	N/A	100	N/A	2	
		7D47	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D48	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D49	(Not Visible)	0	3	0	2	N/A	100	N/A	2	
		7D4A	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D4B	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D4C	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
		7D4D	(Not Visible)	0	0	0	0	N/A	N/A	N/A	2	
Total				0	12	0	7	N/A	100	N/A	34	

As shown in the example on [page 8-4](#), the following statistics are returned:

1. Host time stamp of the sample.
2. Active Symmetrix device name and host physical device name (if the device is visible).
3. Read cache request rate (requests per second).
4. Write cache request rate (requests per second).
5. Read throughput rate (kilobytes per second).
6. Write throughput rate (kilobytes per second).
7. Read cache hit ratio (percentage of read hits).
8. Write cache hit ratio (percentage of write hits).
9. Sequential read ratio (percentage of sequential reads).
10. Write-pending track count.

Device statistics

Using `symstat` with the REQUESTS type (default) specified, you can display performance statistics such as I/O requests via SAs (or other front-end adapters) and throughput on a device.

From the device list shown in the device group statistical run, select a logical device to examine I/O requests and throughput activity and choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 on logical device DEV001 in device group `prod_r1`, enter:

```
symstat -i 5 -c 3 -g prod_r1 -ld DEV001
```

The the fields returned are identical to those returned in “REQUESTS statistics” on [page 8-4](#), except that the statistics are limited to the one logical device specified.

SRDF/A session statistics

Using `symstat` with the REQUESTS type (default) specified, you can display a table of host writes and SRDF throughput information for SRDF/A sessions in Symmetrix arrays by specifying the `-repType rdFa` option.

When requesting SRDF/A session statistics for a specified Symmetrix array the output differs depending upon the flag arguments specified:

- ◆ If the `-sid` flag is used with no further qualifiers (`[ -g ALL | -rdFg <#|ALL> | -cg <CgName|ALL> ]`), then the output display lists SRDF/A sessions that have a status of active.
- ◆ If the `-sid` flag is used with qualifier `-g all`, then the output display list all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdFg ALL`, then the output display lists all RDF groups, regardless of status. If the `-sid` flag is used with the qualifier `-cg ALL`, then the output display lists all composite groups, regardless of status.
- ◆ If the `-g DgName` device group qualifier is used (whereby the `-sid` option is not specified), then the output display only lists details for the specified device group.
- ◆ If the `-sid` flag is used with the qualifier `-rdFg #`, then the output display lists just the specified RDF group.

- ◆ If the `-sid` flag is used with the qualifier `-cg CgName`, then the output display lists just the specified composit group.
- ◆ If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: All device groups

For example, to return SRDF/A session statistics on Symmetrix array 0011 for all active device groups, with a sample intervals of 30 seconds and a sample count of 3, enter:

```
symstat -i 30 -c 3 -sid 0011 -ReptType rdfa
```

The following output is returned. Notice that this option displays rates calculated over the specified interval, therefore more than one interval must be run for the data to be meaningful. The first set of data has no values, since it is a baseline for the remainder statistical data:

```
SRDF/A Session Throughput Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:23:30
```

Device Group Name	RA Grp	Session Number	Host Write Workload		RDF Session Throughput	
			IO/sec	MB/sec	IO/sec	MB/sec
RaGrpNum_25	25	24	0	0.0	0	0.0
RaGrpNum_26	26	25	0	0.0	0	0.0
RaGrpNum_27	27	26	0	0.0	0	0.0

```
SRDF/A Session Throughput Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:24:00
```

Device Group Name	RA Grp	Session Number	Host Write Workload		RDF Session Throughput	
			IO/sec	MB/sec	IO/sec	MB/sec
1	2	3	4	5	6	7
RaGrpNum_25	25	24	353	2.8	691	7.2
RaGrpNum_26	26	25	553	4.3	1004	11.6
RaGrpNum_27	27	26	383	3.0	758	8.1

```
SRDF/A Session Throughput Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:24:30
```

Device Group Name	RA Grp	Session Number	Host Write Workload		RDF Session Throughput	
			IO/sec	MB/sec	IO/sec	MB/sec
RaGrpNum_25	25	24	343	2.7	549	5.7
RaGrpNum_26	26	25	519	4.1	531	6.0
RaGrpNum_27	27	26	398	3.1	741	7.9

The following statistics are returned:

1. Name of the device group.
2. The RA Group number.
3. SRDF/A session number.
4. Number of host I/O writes per second.
5. Workload rate, in MB per second of host writes.
6. Number of SRDF I/O writes per second in the SRDF/A session.

7. Throughput rate, in MB per second of the total I/O writes in the SRDF/A session.

Example: Verbose SRDF/A session statistics

For example, to return verbose details about SRDF/A session statistics on Symmetrix array 6163 for all active device groups, with a sample intervals of 30 seconds and a sample count of 3, enter:

```
symstat -sid 6163 -i 30 -c 3 -reptype rdfa -v
```

SRDF/A Session Information:

```
Symmetrix Id           : 000000006163
Timestamp              : 16:26:24

System Write Pending Limit      : 236462   (7.22 GB)
Cache Slots available for all SRDF/A sessions : 222274   (6.78 GB)
Total Local Write Pending Count : 0
Total System Write Pending Count : 0

Device Group Name          : RaGrpNum_60
RA Group Number            : 60
Session Type               : RDF1
Session Number             : 59
Session Priority            : 33
Number of Device in Session : 3
Session Status             : Active
Session Attributes         : MSC
Local Active Cycle Number   : 443
Minimum Cycle Time         : 00:00:30
Average Cycle Time         : 00:00:30
Last Cycle Time            : 00:00:30
Time Since Last Cycle Switch : 00:00:14
Time that R2 is behind R1 (RPO) : 00:00:44
Tracks not Committed to the R2 Side : 0
RDF1 Active (Capture) Cycle Size : 0
RDF1 Inactive (Transmit) Cycle Size : 0
RDF2 Active (Receive) Cycle Size : N/A
RDF2 Inactive (Apply) Cycle Size : N/A
Cache Slots in Use         : 0
Percent Available Cache Used : 0.0
Host Write Workload (IO/s)  : 0
Host Write Workload (MB/s)  : 0.0
RDF Throughput R1->R2 (IO/s) : 0
RDF Throughput R1->R2 (MB/s) : 0.0
RDF Throughput R2->R1 (IO/s) : 0
RDF Throughput R2->R1 (MB/s) : 0.0
Last Restore Time (R2 only) : 00:00:00
Average Restore Time (R2 only) : 00:00:00
Total HA writes            : 0
Total HA repeat writes     : 0
```

R-Copy session statistics

Using `symstat` with the `REQUESTS` type (default) specified, you can display a table of host writes and SRDF throughput information for Open Replicator R-Copy sessions in Symmetrix arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
        -RepType rcopy [-name <SessionName>]
```

To return R-Copy session statistics for a Symmetrix array 0003, enter:

```
symstat -sid 0003 -reptype rcopy -i 30 -c 3
```

		RCopy	

Device		KB/sec	
		PULL	PUSH
15:28:45			
15:29:20	0644 (Not Visible)	15189	0
	0645 (Not Visible)	15276	0
	0646 (Not Visible)	15275	0
	0647 (Not Visible)	15383	0

This example shows the KB/sec throughput for an R-Copy hot pull operation.

BACKEND statistics and PREFETCH activity

Statistics for I/O requests via back-end directors, the throughput on the corresponding devices, and prefetch disk activity on back-end directors can be returned via the Solutions Enabler SYMCLI.

Returning BACKEND statistics

Using `symstat` with the BACKEND type specified, you can display performance statistics such as I/O requests via a DA (back-end director) and throughput on a device.

From the device list shown in the device group statistical run, select the logical device for which you want to examine I/O requests and throughput activity. Choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 on logical device DEV001 in device group `prod_r1`, enter:

```
symstat -type BACKEND -i 5 -c 3 -g prod_r1 -ld DEV001
```

The display would be similar to the front-end REQUESTS display showing read/write I/O, prefetched tracks, and unused track activity.

PREFETCH activity

From the Symmetrix list, select the Symmetrix array that you want to examine for prefetch disk activity on the back-end directors (DAs). Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 4 for prefetch activity on Symmetrix 010050010333, enter:

```
symstat -i 5 -c 4 -sid 333 -da all -type prefetch
```

The following is an example output:

```
symstat -i 5 -c 4 -sid 333 -da all -type prefetch
```

	DIRECTOR	Prefetch Trk/sec	Prefetch Used	Trk/sec Unused
16:42:30	① ② DA-2A	③ 1	④ 1	⑤ 0
16:42:35	DA-2A	7	6	1
16:42:40	DA-2A	19	19	0
16:42:45	DA-2A	-----	-----	-----
Total		27	26	1

The following statistics are returned:

1. Host time stamp of sample.
2. Active director name type and number.
3. Track prefetch rate (tracks per second).
4. Used prefetched track rate (tracks per second).
5. Unused (superseded) prefetched track rate (tracks per second).

Director statistics

Using `symstat`, you can display the I/O requests and throughput activity performance statistics on any or all directors. In addition, you can return director statistics for R-Copy sessions. To return a list of directors available on all visible Symmetrix arrays, use `symcfg list -dir all`.

Director I/O requests

From the list, select the Symmetrix array for which you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 (for I/O requests and throughput) on Symmetrix 010050010333, enter:

```
symstat -i 5 -c 3 -dir all -sid 097
```

	DIRECTOR	IO/sec	Cache READ	Requests/sec WRITE	Requests/sec RW	% RW Hits
11:09:01						
1	2	3	4	5	6	7
11:09:12	DF-1A	1	0	1	1	0
	FA-3A	108	0	0	0	N/A
	DF-15A	1	0	0	0	N/A
	FA-3B	16	0	3	3	100
	RF-13D	46	0	1	1	0
		-----	-----	-----	-----	---
Total		172	0	5	5	60
11:09:23	FA-3A	114	0	0	0	N/A
	FA-3B	16	0	3	3	100
	RF-13D	43	0	0	0	N/A
		-----	-----	-----	-----	---
Total		173	0	3	3	100

The following statistics are returned:

1. Host time stamp of sample.
2. Active director identifier and number.
3. I/O rates (operations per second) for:
 - Host to Front-end directors
 - Disk to Back-end directors
 - Local to Remote link directors
4. Read cache request rate (requests per second).
5. Write cache request rate (requests per second).
6. Total read/write cache request rate (requests per second).
7. Cache hit ratio (percentage of cache read and write hits).

R-Copy session statistics

Using `symstat` with the `REQUESTS` type (default) specified, you can display a table of directors on the specified array and SRDF throughput information for Open Replicator R-Copy sessions in Symmetrix arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
        -RepType rcopy [-name <SessionName>]
```

To return R-Copy session statistics for a Symmetrix array 097, enter:

```
symstat -sid 097 -reptype rcopy -dir 7a -i 5
```

	Director	RCopy MB/sec	Cache READ	Requests/sec WRITE	RW	%RW Hits
1	2	3	4	5	6	7
16:06:49						
	FA-7A	58	716	0	716	80
		-----	-----	-----	-----	---
		58	716	0	716	79

The following statistics are returned:

1. Host time stamp of sample.
2. Active director identifier and number.
3. R-Copy director throughput in MB/sec.
4. Number, or the count, of cache read requests per second.
5. Number, or the count, of write requests per second.
6. Total number, or count, of combined read and write requests per second.
7. Cache hit ratio (percentage of cache read and write hits).

PORT statistics

Using `symstat` with the `PORT` type specified, you can display performance statistics such as I/O requests and throughput of director ports for a specific Symmetrix array.

How to return director port statistics

Choose a Symmetrix array for which you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run. The `symstat -type PORT` syntax allows you to return statistics for one or all directors on one or all Symmetrix arrays. In addition, you can optionally specify one or all port numbers. Use the following syntax for director port statistics:

```
symstat [-i Interval] [-c Count] -type PORT [-sid SymmID]
        -dir <#|ALL> [-port <#|ALL>]
```

For example, for sample intervals of 5 seconds, and a sample count of 3 (for I/O requests and throughput) on Symmetrix 0256, enter (the output follows):

```
symstat -i 5 -c 5 -type PORT -sid 0256 -dir ALL
```

15:24:30	DIRECTOR	PORT	IO/sec	Kbytes/sec
15:24:36	FA-15C	0	40	0
15:24:41	FA-15C	0	28	0
15:24:47	FA-15C	0	27	0
15:24:52	DIRECTOR	PORT	IO/sec	Kbytes/sec
15:24:52	FA-15C	0	24	0

R-Copy session director port statistics

Using `symstat` with the `PORT` type specified and the `-reptype` set to `rcopy`, you can display a table of director ports on the specified array and session throughput information for Open Replicator R-Copy sessions.

```
symstat -sid 0003 -reptype rcopy -type port -dir 14c -i 30 -c 3
```

15:34:07	Director Port	RCopy		
		MB/sec	Ceiling	Devices
1	2	3	4	5
	FA-14C:1	23	30	4
	FA-14C:1	24	30	4

The following statistics are returned:

1. Host time stamp of sample.
2. Active director and port number.
3. R-Copy session throughput in MB/sec.
4. The maximum percentage of that director/port's bandwidth that is allotted to R-Copy operations.
5. Number of remote devices. The number of remote devices is only available when all R-Copy sessions are non-incremental.

MEMIO statistics

Using `symstat`, you can display performance statistics, such as cache memory to disk I/O on a device group. But first, you may need to list the current device groups (use `symdg list`). From the device group list, select the device group for which you want to examine the I/O or cache memory to disk activity. Choose the interval and number of samples (counts) for the statistical run.

Example: Memory and I/O statistics

For example, to examine cache to disk activity on device group `prod_r1` with a sample intervals of 5 seconds and a sample count of 3, enter:

```
symstat -i 5 -c 3 -g prod_r1 -type memio
```

By default, BCV devices are not included in these results. Optionally, you can choose to return statistics for only BCV devices in the device group by specifying the `-bcv` option. Alternatively, you can choose to return statistics for both standard and BCV devices in the device group by specifying the `-all` option. The following display is an example output of a device group statistical run showing cache to disk activity for the devices:

```
symdg list
```

D E V I C E		G R O U P S			
Name	Type	Symmetrix ID	Num of Devices	Num of GateKeepers	Num of BCV's
prod_r1	RDF1	010010020233	5	0	0
prod_r2	RDF1	010010020488	48	0	0

```
symstat -i 5 -c 3 -g prod_r1 -type memio
```

DEVICE		Tracks/sec			% Dev
1	2	3	4	5	6
WP	Tracks	Prefchd	Destgd		WPmax
15:13:32					
15:13:37	DEV001 (0006)	8	0	51	0
	DEV002 (0007)	15	0	51	0
	DEV003 (0009)	35	0	51	0
	DEV004 (000A)	15	0	51	0
	DEV005 (000B)	25	0	51	0
	-----	98	0	255	0
15:13:42	DEV001 (0006)	21	0	51	0
	DEV002 (0007)	20	0	51	0
	DEV003 (0009)	15	0	102	0
	DEV004 (000A)	23	0	51	0
	DEV005 (000B)	11	0	102	0
	-----	90	0	357	0

The following statistics are returned:

1. Host time stamp of sample.
2. Active logical device name and Symmetrix device name.
3. Write pending track count not yet destaged to disk.
4. Track prefetch rate (tracks per second).
5. Track destage rate (tracks per second).
6. Write-pending device limit percentage (percentage of device parameter).

DMSP statistics

Using `symstat` with the DMSP type specified, you can display DMSP (dynamic mirroring service policy) statistics such as service policy and I/O performance attributes for a specified Symmetrix device.

Device dynamic mirroring service policy statistics

From the device list shown in the device group statistical run, select the logical device for which you want to examine DMSP statistics and throughput activity. Choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 on logical device `DEV001` in device group `prod_r1`, enter:

```
symstat -type DMSP -i 5 -c 3 -g prod_r1 -ld DEV001
```

The display provides DMSP information for mirror 1 and 2, the preferred mirror, new and old mirror service policies, the percent of sequential I/O, the percent of write I/O, and the activity level.

Disk statistics

Using `symstat` with the `DISK` type specified, you can display performance statistics, such as I/O requests and throughput on a physical disk, at specified time intervals for the duration of a specified count.

Specifying a disk

Select the physical DA director number, interface ID, and disk SCSI ID interface on a specified Symmetrix array for which you want to examine I/O requests and throughput activity. To retrieve the statistics of all DA directors, interfaces, or disk SCSI IDs, use the value `ALL`. For example, for sample intervals of 5 seconds and a sample count of 3 on disk `02A:C5`, enter:

```
symstat -type disk -i 5 -c 3 -sid 577 -disk 2a,C,5
```

If you want to retrieve statistics at the same interval and sample count, but for all SCSI IDs on DA 02A, interface C enter:

```
symstat -type disk -i 5 -c 3 -sid 577 -disk 2a,C,ALL
```

The display provides the disk ID, I/O read/writes, and kilobytes per read and write for each disk.

CACHE statistics

You can return data about the cache activity on the front-end directors, LRU, or SRDF/A sessions.

Front-end director statistics

Statistics for front-end directors can be returned by specifying options for SA, EA, FA, FE, CA, RA, director number, or all. In addition you can choose the interval and number of samples (counts) for the statistical run. For example, to return cache activity for all SAs on Symmetrix 010050010333, for sample intervals of 5 seconds and a sample count of 4, enter:

```
symstat -i 5 -c 4 -sid 097 -sa all -type cache
```

	DIRECTOR	Misses/sec Cache Read	Disconnects/sec System	Device
15:21:52				
15:22:03	FA-15C	0	24	0

As shown in this example, there was one sample of cache activity to report on one director during the sample period and `symstat` lists the following statistical items:

1. Host time stamp of sample.
2. Active director identifier and number.
3. Cache read miss rate (misses per second).
4. System disconnect rate (disconnects per second) due to write pending limit.
5. Device disconnect rate (disconnects per second) due to device write pending limit.

LRU cache management statistics

Using `symstat`, you can display the LRU cache statistics. But first, you may need to list the LRUs and Symmetrix arrays (use `symcfg list -lru ALL`) to see the LRUs for all units and you can also specify a Symmetrix ID.

For example, to obtain a list of ALL LRUs on Symmetrix 000184600063, and their cache partitioning, enter:

```
symcfg list -lru ALL -sid 063
```

The following is an example output:

LRU Num	LRU Name	Cache Slots	Percent of Total
1	LRU_0	13831	25%
2	LRU_1	13831	25%
3	LRU_2	13831	25%
4	LRU_3	13831	25%
Total		55325	

Select the Symmetrix array for which you want to examine the LRUs I/O requests and throughput activity. Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 (for I/O requests and throughput) on Symmetrix 0011, enter:

```
symstat -type cache -lru all -i 5 -c 3 -sid 0011
```

The following display is an example of the LRU cache statistics obtained using this command:

	#	LRU NAME	Cache Requests/sec		RW	% RW HITS	Locks/sec
			READ	WRITE			
14:29:44	1	GROUP_0	5353	835	6189	75	0
14:30:09	1	GROUP_0	3468	537	4005	75	0

The following statistics are returned:

1. Host time stamp of sample.
2. LRU number.
3. LRU name.
4. Read cache request rate (requests per second).
5. Write cache request rate (requests per second).
6. Total read/write cache request rate (requests per second).
7. Cache hit ratio (percentage of cache read and write hits).
8. LRU lock request rate per second.

SRDF/A session statistics

Use the `symstat` command to return a table of cache usage for device groups (SRDF/A sessions) in the specified Symmetrix that have active status or have non zero cache usage by specifying the `-repType rdfa` option. SRDF/A cache statistic data can be returned for all or a specific device group or RA group number.

When requesting SRDF/A session statistics for a specified Symmetrix array the output differs depending upon the flag arguments specified:

- ◆ If the `-sid` flag is used with no further qualifiers (`[ -g ALL | -rdfg <#|ALL> | -cg <CgName|ALL> ]`), then the output display lists SRDF/A sessions that have a status of active.
- ◆ If the `-sid` flag is used with qualifier `-g all`, then the output display list all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg ALL`, then the output display lists all RDF groups, regardless of status. If the `-sid` flag is used with the qualifier `-cg ALL`, then the output display lists all composite groups, regardless of status.
- ◆ If the `-g DgName` device group qualifier is used (whereby the `-sid` option is not specified), then the output display only lists details for the specified device group.
- ◆ If the `-sid` flag is used with the qualifier `-rdfg #`, then the output display lists just the specified RDF group.
- ◆ If the `-sid` flag is used with the qualifier `-cg CgName`, then the output display lists just the specified composite group.
- ◆ If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

**Example: R1 SRDF/A
session statistics for
active groups**

For example to return SRDF/A session statistics for all active device groups on Symmetrix array 0011, enter:

```
symstat -type cache -i 30 -c 3 -sid 0011 -repType rdfa
```

The following statistics are returned:

SRDF/A Session Cache Summary Information

```
Symmetrix Id      : 000187400011
Timestamp         : 14:32:31
System Write Pending Limit : 156073 (4.76 GB)
Cache Slots available for all SRDF/A sessions : 146708 (4.48 GB)
Total Local Write Pending Count : 67332
Total System Write Pending Count : 69340
```

Device Group Name	Session					Cache Slots In Use	%Available Cache Used
	RA Grp	Type	Number	Priority	Status		
RaGrpNum_25	25	RDF1	24	33	Active	1749	1.2
RaGrpNum_26	26	RDF1	25	33	Active	2722	1.9
RaGrpNum_27	27	RDF1	26	33	Active	2018	1.4
Total						6489	4.4
Slots						0.20	
GB							

SRDF/A Session Cache Summary Information

```
Symmetrix Id      : 000187400011
Timestamp         : 14:33:01
System Write Pending Limit : 156073 (4.76 GB)
Cache Slots available for all SRDF/A sessions : 146708 (4.48 GB)
Total Local Write Pending Count : 67390
Total System Write Pending Count : 71576
```

Device Group Name	Session					Cache Slots In Use	%Available Cache Used
	RA Grp	Type	Number	Priority	Status		
RaGrpNum_25	25	RDF1	24	33	Active	3089	2.1
RaGrpNum_26	26	RDF1	25	33	Active	4913	3.3
RaGrpNum_27	27	RDF1	26	33	Active	2953	2.0
Total						10955	7.5
Slots						0.33	
GB							

SRDF/A Session Cache Summary Information

```
Symmetrix Id      : 000187400011
Timestamp         : 14:33:31
System Write Pending Limit : 156073 (4.76 GB)
Cache Slots available for all SRDF/A sessions : 146708 (4.48 GB)
Total Local Write Pending Count : 67291
Total System Write Pending Count : 69722
```

Device Group Name	Session					Cache Slots In Use	%Available Cache Used
	RA Grp	Type	Number	Priority	Status		
RaGrpNum_25	25	RDF1	24	33	Active	1894	1.3
RaGrpNum_26	26	RDF1	25	33	Active	2944	2.0
RaGrpNum_27	27	RDF1	26	33	Active	2254	1.5
Total						7092	4.8
Slots						0.22	
GB							

The following statistics are returned:

- ◆ **System Write Pending Limit** — The number of cache slots available in the Symmetrix array, which are used to hold writes awaiting destaging locally or transfer in a SRDF/A cycle. Typically, this limit is approximately 80% of the total cache available, however, in some case it is less.
- ◆ **Total Local Write Pending Count** — Count of cache slots with writes pending to a local device. This count is a device level statistic, representing the total value for all devices in a given Symmetrix array.
- ◆ **Total System Write Pending Count** — Count of cache slots in the Symmetrix array with a write pending flag. Cache slots can have either writes pending to a local device, writes pending to a SRDF/A device, or both. When a cache slot has both local and SRDF/A writes pending flag set, it is only counted once.
- ◆ **Cache Slots Available for all SRDF/A sessions** — The upper limit of cache slots available for all SRDF/A sessions. This value typically is determined by the SRDF/A Max Cache Usage parameter, which allows the user to specify what percentage of the System Write Pending Limit slot count should be used for SRDF/A sessions. If the Max Host Throttle parameter is 0, then Cache slots available for all SRDF/A sessions is a percentage of System Write Pending Limit as specified by SRDF/A Max Cache Usage setting. The default value of this limit is approximately 94% of the System Write Pending limit. If the Max Host Throttle parameter is not 0, then the number of Cache Slots Available for all SRDF/A sessions will be equal to the System Write Pending limit.
- ◆ **Cache Slots in Use** — Cache slots in use is the total number of cache slots in the active and inactive cycle for that SRDF/A session. These represent any cache slot with a SRDF/A write pending.

Note: The sum of the Cache Slots in Use and the Local Write Pending Count does not equal the Total System Write Pending Count. This is because Cache Slots in Use and Local Write Pending Count include every cache slot in those states, while the Total System Write Pending Count includes slots with either of those states, but not both. So a cache slot with both a local write pending and a SRDF/A write pending would increase the individual counts of each, but only increment the total count by one.

- ◆ **%Available Cache Used** — The percentage of the available SRDF/A cache in use for the device group.

PATH statistics

Using the SYMCLI you can report R-Copy path information for non-incremental sessions. Symmetrix arrays that have all or some incremental sessions will report an error.

Note: These statistics are only available when all R-Copy sessions are non-incremental.

The data can either be presented by session, where the data is organized by session name (-by_session option), or by port, where the data will be organized by director port (-by_port option).

Organized by session The following example shows the R-Copy PATH statistics for Symmetrix array 003 organized by session name:

```
symstat -sid 0003 -reptype rcopy -type path -by_session -i 30 -c 3
```

Time Stamp	Session Name	Control Device	Target Destination	Tracks Total	Tracks Remaining	Director Port	RCopy Ceiling
15:43:24	hotpull	0644	N/A	264960	0	FA-14C:1	30
	hotpull	0645	N/A	264960	0	FA-14C:1	30
	hotpull	0646	N/A	264960	0	FA-14C:1	30
	hotpull	0647	N/A	264960	0	FA-14C:1	30

Time Stamp	Session Name	Control Device	Target Destination	Tracks Total	Tracks Remaining	Director Port	RCopy Ceiling
15:44:00	hotpull	0644	N/A	264960	0	FA-14C:1	30
	hotpull	0645	N/A	264960	0	FA-14C:1	30
	hotpull	0646	N/A	264960	0	FA-14C:1	30
	hotpull	0647	N/A	264960	0	FA-14C:1	30

Time Stamp	Session Name	Control Device	Target Destination	Tracks Total	Tracks Remaining	Director Port	RCopy Ceiling
15:44:36	hotpull	0644	N/A	264960	0	FA-14C:1	30
	hotpull	0645	N/A	264960	0	FA-14C:1	30
	hotpull	0646	N/A	264960	0	FA-14C:1	30
	hotpull	0647	N/A	264960	0	FA-14C:1	30

Organized by port The following example shows the R-Copy PATH statistics for Symmetrix array 003 organized by director port:

```
symstat -sid 0003 -reptype rcopy -type path -by_port -i 30 -c 3
```

Time Stamp	Director Port	RCopy Ceiling	Port KB/sec	RCopy KB/sec	Control Device	Tracks Total Remaining	Session Name
15:44:44	FA-14C:1	30	-	-	0644	264960	0 hotpull
					0645	264960	0 hotpull
					0646	264960	0 hotpull
					0647	264960	0 hotpull

Time Stamp	Director Port	RCopy Ceiling	Port KB/sec	RCopy KB/sec	Control Device	Tracks Total Remaining	Session Name
15:45:20	FA-14C:1	30	17	0	0644	264960	0 hotpull
					0645	264960	0 hotpull
					0646	264960	0 hotpull
					0647	264960	0 hotpull

Time Stamp	Director Port	RCopy Ceiling	Port KB/sec	RCopy KB/sec	Control Device	Tracks Total Remaining	Session Name
15:45:55	FA-14C:1	30	19	0	0644	264960	0 hotpull
					0645	264960	0 hotpull
					0646	264960	0 hotpull
					0647	264960	0 hotpull

CYCLE statistics

The CYCLE type statistics return a table of cycle information for device groups (SRDF/A sessions) in Symmetrix arrays that have active status or have non zero cache usage.

When requesting SRDF/A session statistics for a specified Symmetrix array the output differs depending upon the flag arguments specified:

- ◆ If the `-sid` flag is used with no further qualifiers (`-g all, -rdfg # | ALL`), then the output display lists SRDF/A sessions that have a status of active.
- ◆ If the `-sid` flag is used with qualifier `-g all`, then the output display lists all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg all`, then the output display lists all RDF groups, regardless of status.
- ◆ If the `-sid` flag is used with the qualifier `-rdfg #`, then the output display lists just the specified RDF group.
- ◆ If the `-g DgName` device group qualifier is used (without specifying the `-sid`), then the output display only lists details for the specified device group.
- ◆ If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: All device groups

For example, to return SRDF/A session cycle statistics of active sessions for Symmetrix array 0011, enter:

```
symstat -type cycle -i 30 -c 3 -sid 0011 -RepType rdfa -g all
```

The following statistics are returned for one count:

```
SRDF/A Session Cycle Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:40:16
```

		Session			Cycle Time (sec)				Cycle Size		
		Type	Number	Active Cycle#	Min	Avg	Last	Switch	Active	Inactive	
Device	Group Name	RA Grp									
1		2	3	4	5	6	7	8	9	10	11
< . . . >											
RaGrpNum_25		25	RDF1	24	171	15	19	20	4	732	526
RaGrpNum_26		26	RDF1	25	165	15	19	22	2	591	4221
RaGrpNum_14		14	RDF1	13	173	15	19	19	4	0	0
RaGrpNum_16		16	RDF1	15	173	15	19	20	4	0	0
RaGrpNum_18		18	RDF1	17	172	15	19	19	4	0	0
RaGrpNum_19		19	RDF1	18	177	15	19	19	4	0	0
RaGrpNum_20		20	RDF1	19	171	15	19	19	4	0	0
RaGrpNum_27		27	RDF1	26	165	15	19	21	2	426	1294

Legend for the Attribute of Cycle Size:

```
RDF1: Active = Capture   Inactive = Transmit
RDF2: Active = Receive   Inactive = Apply
```

Note: Note: The table headings Active and Inactive are used to represent the Captured and transmitted data on the R1 side and Receive and Apply on the R2 side of the operation respectively.

The following statistics are returned:

1. Device Group Name.
2. RA Group number.

Session

3. Type of SRDF group; inactive sessions do not have an assigned type.
4. SRDF/A session number.
5. Active cycle number identifier for the given SRDF/A session. MSC cycle numbers are not displayed.

Cycle Time

6. Min — The setting for the minimum number of seconds for a specific cycle.

Note: The Min Cycle Time value is not shown on the R2 side.

7. Avg — A measure of the average time of all cycles in the session in seconds.
8. Last — A measure of the last cycle in seconds.
9. Last Switch — A measure of time in seconds since the last switch from active to inactive or vice versa.

Cycle Size

10. Active — A count of cache slots that measures the amount of data captured (on the R1 side) or received (on the R2 side).
11. Inactive — A count of cache slots that measures the amount of data transmitted (on the R1 side) or applied (on the R2 side).

This chapter introduces the Change Tracker SYMCLI command (`symchg`) command and explains how to monitor Symmetrix data objects for change within your storage environment.

- ◆ Introduction to change tracker9-2
- ◆ Monitoring data objects for changes.....9-4
- ◆ Generating reports of changes.....9-9
- ◆ Using Change Tracker for RDF capacity planning9-13

Introduction to change tracker

The EMC Solutions Enabler Change Tracker software is used to measure changes to data on a Symmetrix volume or group of volumes. The Change Tracker command (`symchg`) is used on FBA devices to mark selected areas of Symmetrix disk storage so that Symmetrix logical volume occupying those areas can be monitored for change.

Change Tracker uses DeltaMark bitmap technology (formerly called the Symmetrix Differential Data Facility) to identify logical blocks that have been changed on a Symmetrix FBA device. Before change tracking can begin, a Change Tracker (DeltaMark) session must be created using the `symchg create` command. The `symchg mark` command is then used to perform a timestamp and mark the selected area of disk storage occupied by a data object using a DeltaMark bitmap.

This allows a marked area to be examined at a later time to view changes to the stored object. The tracked changes can also be saved to a log file for later analysis. Change Tracker data is often used to analyze and design TimeFinder and Symmetrix Remote Data Facility (SRDF) configurations.

The Symmetrix system stores information about track changes internally, and makes this information available to applications through a DeltaMark bitmap vector or table. Each DeltaMark bitmap contains a bit for every track on the monitored volume. When a write to the volume alters a track of information, the bitmap is updated to reflect that the track has been changed. (Subsequent changes to that same track have no effect on the bitmap.)

Figure 9-1 illustrates a DeltaMark bitmap table.

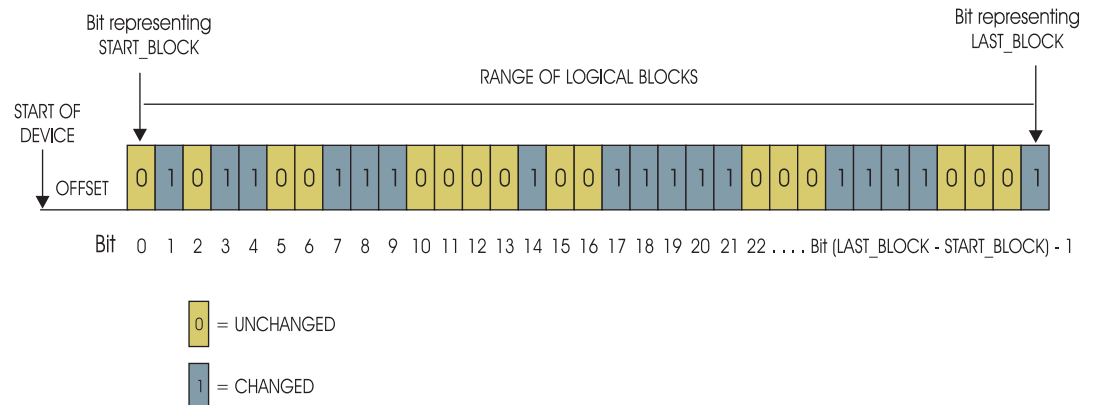


Figure 9-1 A DeltaMark bitmap table

Practical uses

Using TimeFinder and SRDF software to mirror data from a source Symmetrix system to a remotely associated Symmetrix system located across the hall or across the globe creates a global data storage solution. That is, production data can be available at a remote site for data mining, e-business content delivery, application development, testing, backups, and numerous other uses. Moreover, SRDF configurations provide a recovery solution for component or site failures between remotely mirrored devices.

With the wide acceptance of SRDF for remote storage and as a disaster tolerance and disaster restart tool, assessing incremental changes to data over discrete periods of time is a necessary part of planning for data transfer and storage at remote sites. Change Tracker assists in this task by measuring volume changes, providing information so that informed decisions can be made about the communication line bandwidths and links required for such configurations.

Change Tracker command summary

The Change Tracker command (`symchg`) allows you to monitor selected Symmetrix disk storage objects using DeltaMark bitmap technology to track changes and generate reports for later analysis. Symmetrix objects that can be monitored include individual devices, standard or BCV devices listed in a device group or composite group, and devices listed in a device file.

[Table 9-1](#) summarizes the SYMCLI `symchg` commands you can use to create a Symmetrix Change Tracker session, mark areas to be monitored, remove or delete a session, and generate a report.

Table 9-1 Change Tracker command summary

Command	Argument	Displays
<code>symchg</code>	<code>create</code>	Creates a Change Tracker session for the disk device containing the disk storage object to be monitored.
	<code>delete</code>	Unmarks the disk storage object and deletes the Change Tracker session.
	<code>list</code>	Lists all disk storage objects that are currently marked or all devices that have a Change Tracker session created.
	<code>mark</code>	Marks disk storage objects as unchanged so that new changes can be tracked. Requires that a Change Tracker session first be created.
	<code>remove</code>	Unmarks the disk storage object while retaining the Change Tracker session.
	<code>report</code>	Generates a report on the amount of change for the specified object in a given log file. Reports can be generated for device groups, composite groups, and device files.
	<code>view</code>	Displays the amount and rate of change for marked disk storage objects or shows if a Change Tracker session exists for a specified disk storage object.

For information about performing Change Tracker control operations, refer to [“Monitoring data objects for changes” on page 9-4](#). For details about the `symchg` command syntax and other SYMCLI commands, refer to *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Note: Change Tracker control operations using the `symchg` command require that a DeltaMark license key is installed on your system. Refer to the *EMC Solutions Enabler Installation Guide* for license information.

Monitoring data objects for changes

To mark areas of Symmetrix disk storage for change monitoring, you first create a Change Tracker session for the objects that you want to monitor. Creating a Change Tracker session creates DeltaMark bitmaps for the specified objects. Once a session has been established, the objects can be marked at any particular point in time, which resets the bitmaps for those objects (that is, marks all tracks as unchanged). Marking the objects is the starting point from which changes will be measured.

Monitoring the objects is accomplished by viewing the objects in a session. Viewing objects processes the bitmaps that correspond to the objects and displays the amount and rate of change. Viewing can be performed at discrete intervals of time and for a particular number of executions, and the bitmaps can also be reset after each sample. The display may be saved to a user-defined log file, where it is stored in a comma separated format, enabling the data to be imported into a spreadsheet for analysis. A summary report can be produced from the results in a log file, and the log file can be used to perform capacity planning by specifying particular variables and calculating other parameters.

The `symchg` command operates on Symmetrix disk storage objects by specifying a Symmetrix device name, a logical device name, a device group, a composite group, or a device file. Individual objects may be removed from a session without terminating that session (so other objects in the same session continue to be monitored), and a session can be deleted, terminating the monitoring of all objects included in that session.

Note: SYMCLI `symchg` only monitors FBA devices. CKD devices can be monitored using the mainframe change tracker utility.

Creating a Change Tracker session

Before marking objects, you must create a Change Tracker session for the device(s) containing the objects by using the `symchg create` command.

The `symchg create` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName create
symchg -g DgName create
symchg -cg CgName create
symchg -file FileName create
```

For example, to create a Change Tracker session for a single Symmetrix device (17c) on Symmetrix ID 000125600055, enter:

```
symchg create dev 17c -sid 55
```

To create a Change Tracker session for devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB create
```

To create a Change Tracker session for devices listed in a device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 create
```

You can mark and examine more than one object per session and you can specify a Symmetrix device name (*SymDevName*) or a logical device (*LdevName*) name within a device group or composite group for the session. You can specify the standard or BCV device (`-bcv`) of a device group or composite group.

Creating a device text file

You can create a Change Tracker session for devices listed in a device text file. The device file syntax contains a single column of devices specified by Symmetrix device name (*SymDevName*). Lines in the device file that begin with a pound symbol (#) will be ignored. The device filename (`-file FileName`) will be inserted into the command line for control operations. A sample of a device text file listing Symmetrix devices is shown in the following example.

Example of a device text file

```
# dev_file_1
# Symmetrix Device List for Change Tracker session
#
00A
00B
00C
057
058
# End
```

Marking a data object

After creating a session, you can mark the device storage objects for change monitoring using the `symchg mark` command. This allows objects to be marked from a particular point in time, with all tracks being marked as unchanged. Once marked, the objects can be monitored for changes.

The `symchg mark` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName mark
symchg -g DgName mark
symchg -cg CgName mark
symchg -file FileName mark
```

Note: You must use the same convention for the mark command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

When you mark an object, an entry is added or updated for the specified object in the `symchg` database. The mark command causes the DeltaMark bitmaps to be reset (no changes) for the devices of the specified object.

You can specify a Symmetrix device name (00A) or a logical device name (DEV001) for the object to be marked. You can also mark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to mark Symmetrix device (17C) on Symmetrix ID 000125600055, enter:

```
symchg mark dev 17c -sid 55
```

To mark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB mark -bcv
```

For example, to mark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 mark
```

Listing marked objects

You can use the `symchg list` command to list disk storage objects that have been marked.

For example, to list all logical disk storage objects that are currently marked, enter:

```
symchg list
```

To list all disk devices for Change Tracker sessions that have been created, enter:

```
symchg list -session
```

To list all disk devices that are visible to the host for which Change Tracker sessions that have been created, enter:

```
symchg list -session dev
```

The following is an example of the `symchg list` command output for Symmetrix 000125600055.

```
symchg -sid 55 -session list
```

Symmetrix ID: 000125600055								
CHANGE TRACKER DELTAMARK SESSIONS								
Device Name		Directors			Device			
Physical	Sym	SA	:P	DA	:IT	Config	Attribute	Sts
								Cap (MB)
/dev/rdisk/c3tl3d9s2	108	14A:1	02B:D3			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d10s2	109	14A:1	01A:C3			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d11s2	10A	14A:1	01B:D2			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d12s2	10B	14A:1	02A:C2			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d13s2	10C	14A:1	02B:D1			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d14s2	10D	14A:1	01B:D1			2-Way Mir	Grp'd	RW 4096
/dev/rdisk/c3tl3d15s2	10E	14A:1	02A:D1			2-Way Mir	Grp'd	RW 4096

Removing marked objects

You can unmark a disk storage object from the `symchg` database without terminating the Change Tracker session by using the `symchg remove` command.

The `symchg remove` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName remove
symchg -g DgName remove
symchg -cg CgName remove
symchg -file FileName remove
```

Note: You must use the same convention for the `remove` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

The `remove` command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object, while maintaining the session.

You can specify a Symmetrix device name (i.e. 00A) or a logical device name (i.e. DEV001) for the object to be unmarked. You can also unmark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to unmark Symmetrix device (17C) on Symmetrix ID 000125600055, enter:

```
symchg remove dev 17c -sid 55
```

To unmark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB remove -bcv
```

To unmark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 remove
```

Deleting marked objects

You can unmark a disk storage object from the `symchg` database and delete (terminate) the Change Tracker session by using the `symchg delete` command.

The `symchg delete` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName delete
symchg -g DgName delete
symchg -cg CgName delete
symchg -file FileName delete
```

Note: You must use the same convention for the `delete` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

The `delete` command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object and deletes the session.

You can specify a Symmetrix device name (i.e. 00A) or a logical device name (i.e. DEV001) for the object to be unmarked. You can also unmark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to unmark Symmetrix device (17C) on Symmetrix ID 000125600055 and delete the session, enter:

```
symchg delete dev 17c -sid 55
```

To unmark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB delete -bcv
```

To unmark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 delete
```

Viewing marked objects

You can process the DeltaMark bitmap corresponding to the disk storage object to display the amount and the rate of change by using the `symchg view` command. For example, to display the amount and the rate of change for Symmetrix device (17c) on Symmetrix ID 000125600055, enter:

```
symchg view dev 17c -sid 55
```

Note: You can verify the existence of a Change Tracker session for a specified object by including the `-session` option.

To display the amount and the rate of change for BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB view -bcv
```

To display the amount and the rate of change for devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 view
```

Change information for marked objects in a device group, composite group, or listed in a device file can be saved to a separate log file. For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB`, enter:

```
symchg -g ProdDB view -reset -log /tmp/logfile -c 10
```

The `-reset` option resets the DeltaMark bitmaps for each reporting interval and requires using the count (`-c`) or interval (`-i`) options. The reset after each measurement of changed tracks is called the *DELTA* collection method.

The *SUM* collection method calculates the cumulative number of changed tracks from the starting point of when the object was first marked. To collect changed tracks for both the *SUM* and *DELTA* collection methods at the same time, you can use the `-multi` option.

For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB` using both the *SUM* and *DELTA* collection methods, enter:

```
symchg -g ProdDB view -multi -log /tmp/logfile -c 10
```

Note: At each specified interval, the log file will capture both the cumulative number of changed tracks and the number of changed tracks per interval.

Generating reports of changes

Change Tracker allows you to generate two kinds of reports — *SUM* and *DELTA*. Reports can be generated for devices in a device group, composite group, or devices listed in a device file.

SUM reports

SUM reports track changes from the point-in-time the DeltaMark bitmap was marked at the beginning of change measurement. Subsequent writes to the same track are not monitored, because the bitmap is not reset after every measurement interval. For instance if the initial *mark* action occurred at 6 a.m. and change data was collected each hour for 12 hours, it would be possible to measure the tracks changed between 6 and 7 a.m., 6 and 8 a.m., 6 and 9 a.m. and so on. It would not be possible to measure the changes between 7 and 8, or 8 and 9 using the SUM report. The SUM reports lend themselves to refresh interval estimations for SRDF/AR installations. Understanding the change measurement at various intervals, enables one to determine the amount of SRDF bandwidth required for different SRDF/AR refresh intervals. Refer to the log file entry below.

```
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 59525 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 91946 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d4s2 , 0 , 131070 , SUM
```

A B C D E F G H I

The fields in the log file entry are:

- A - Date and time that the sample was collected.
- B - Symmetrix ID.
- C - Type of entity (device group).
- D - Name of the device group.
- E - Symmetrix device name.
- F - Physical device name. If the device has not been host visible, this field would say *Not Visible*.
- G - Number of tracks changed since the group was marked.
- H - Total number of tracks on device.
- I - SUM indicates that the number in column G represents the cumulative tracks changed since the group was marked.

DELTA reports

DELTA reports track changes that have occurred since the last measurement interval. The DeltaMark bitmap is reset after each measurement of changed tracks. Thus, an hourly report from 6 a.m. to 6 p.m. using the DELTA method would identify how many tracks changed during any given one hour period. This is useful for determining peak change rates for a given interval of data collection.

To assess the bandwidth requirements for an SRDF/AR implementation, you should first collect a SUM report to determine how often you would want to refresh the remote site. Once the determination of the refresh interval has been finalized (T hours), generate a DELTA report using T hours as the collection interval. This will result in identifying the peak number

of changed tracks during any T hour interval. Refer to the log file entry below for an example of a DELTA report.

```
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 16 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d1s2 , 8 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d1s2 , 803 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d1s2 , 549 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 24 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d1s2 , 48 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d1s2 , 1085 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d1s2 , 809 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 108 , /dev/rdisk/c3t13d1s2 , 25 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 109 , /dev/rdisk/c3t13d1s2 , 13 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 10A , /dev/rdisk/c3t13d1s2 , 561 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 10B , /dev/rdisk/c3t13d1s2 , 541 , 131070 , DELTA
```

Viewing reports

When viewing the changes for an object of a device group, the output can either be sent to `stdout` or saved to a user-defined log file. The information saved to the log file is stored in a comma-separated format. This allows the data to be imported to a spreadsheet for analysis. The results in the log file can be examined with the following command:

```
symchg report -log LogFile -g DgName ...
```

Each report in a log file shows the amount of change for a specified object of a device group. For more information about the syntax of `symchg`, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference* manual.

Report options

The following command line options are available for generating a report using the `symchg report` command:

Table 9-2 SYMCLI report options

Option	Description
-backend	Displays information about the DA and backend disk address of marked devices.
-bcv	When used with the report command, generates a report for the BCV devices in the device group or composite group.
-log	When used with the report command, specifies the log file (<code>-log LogFile</code>) to read from.
-start	Defines the start point in the log file to begin processing data. Format is <code>mmddyyyy hh:mm</code> . Default is the beginning of the log file.
-stop	Defines the stop point in the log file to stop processing data. Format is <code>mmddyyyy hh:mm</code> . Default is the end of the log file.
-ra	Defines the number of RA directors to use when resynchronizing device pairs in the group.
-range	Specifies a range of Symmetrix device names (<i>SymDevName</i>) to generate the report for. Format is <i>StartDev:EndDev</i> .
-rate	Specifies the transfer rate to use when resynchronizing device pairs in the group. Format is in <i>Kb/second</i> .
-resync	Specifies the required time frame in minutes or hours and minutes for resynchronizing RDF device pairs in the group. Format is <i>mmm</i> or <i>hh:mm</i> .

Example procedure

The following is an example of typical steps to follow when you are saving change information to a user-defined log file and you are generating a report:

1. Create Change Tracker sessions for each standard device in a device group (such as DATABASE_DG), enter:

```
symchg create -g DATABASE_DG
```

2. Mark the devices within the device group (DATABASE_DG).

```
symchg mark -g DATABASE_DG
```

This will reset the DeltaMark change information to zero for each device in the device group.

3. Save the change information for the devices of the group (DATABASE_DG) to a log file (/tmp/change_data.log) by entering:

For SUM Reports:

```
symchg view -g DATABASE_DG -log /tmp/change_data.log \ -i 60 -c 100
```

For DELTA Reports:

```
symchg view -g DATABASE_DG -log /tmp/change_data.log \ -reset -i 60 -c 100
```

Note: In this example, there were 100 samples made at an interval of 60 seconds and the DeltaMark change information is reset after each sample.

For combined SUM and DELTA Reports:

```
symchg view -g DATABASE_DG -log /tmp/change_data.log -multi -i 60 -c 100
```

4. Produce a summary report of the changed track information for each device in the group (DATABASE_DG) by entering:

```
symchg report -g DATABASE_DG -log /tmp/change_data.log -v
```

The data in the log file is in a common-separated format, so you could also load it into a spreadsheet to view the data.

5. If you need to provide RDF capacity planning that determines how long it would take to resynchronize a device group (DATABASE_DG) with RA directors at a transfer rate of 1000 KB/second, enter:

```
symchg report -g DATABASE_DG -log /tmp/change_data.log -rate 1000 -ra 2
```

6. If you need to view information for a device group (DATABASE_DG) on your screen showing the log entries, enter:

```
symchg view -g DATABASE_DG
```

7. List the objects that are marked and being monitored for change, enter:

```
symchg list
```

8. Verify a Change Tracker session has been created for a device group (DATABASE_DG) and that it is marked, enter:

```
symchg view -g DATABASE_DG -session
```

9. Remove the marked object (DATABASE_DG), enter:

```
symchg remove -g DATABASE_DG
```

This does not delete the sessions for the devices of the group. It simply records the object indicating it is no longer being monitored (from a host view point).

Using Change Tracker for RDF capacity planning

The Symmetrix Remote Data Facility (SRDF) maintains a mirror image of data at the device level in Symmetrix systems located in physically separate sites. The mirroring in SRDF can be accomplished in real time (Synchronous mode) or near real time (Asynchronous or Adaptive Copy modes). These modes of operation are described fully in the *EMC Solutions Enabler Symmetrix SRDF Family Product Guide*. The Adaptive Copy mode is especially useful when transferring large amounts of data during data center migrations, consolidations, and in data mobility environments.

When performing periodic RDF transfers using Adaptive Copy mode, it is important to estimate the rate of change of data on the volumes to be mirrored in order to make informed decisions about communication line parameters. Change Tracker can help in this process by calculating certain RDF capacity values (the resynchronization time, the number of remote adapters (RAs), and the RA transfer rates) based on the amount of change to monitored objects.

In the SRDF configuration shown in [Figure 9-2](#), the R1-BCV is used to periodically transmit changed tracks from the primary site to the remote site. When the BCV is established with the standard as a BCV pair, the RDF link to its R2 target is suspended. When you split the local BCV pair, you can resume the RDF link, sending the changes to the remote site. Using Change Tracker to measure the extent of changes to the local BCV pair while the R1-BCV is suspended from its R2 target allows you to make informed choices about link bandwidth and frequency of transmission for the RDF transfer.

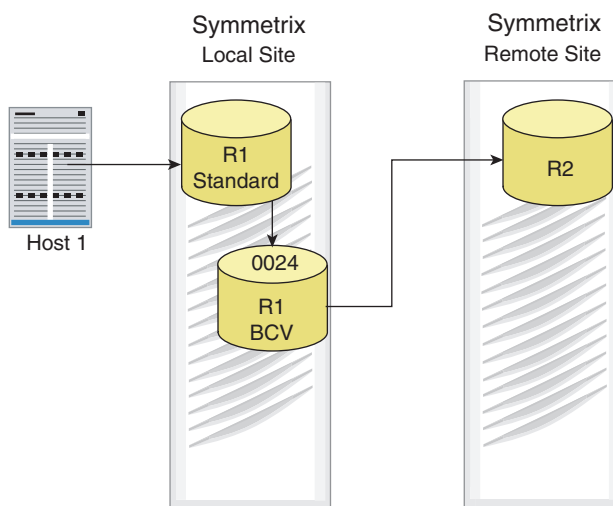


Figure 9-2 SRDF configuration

The following steps outline an example of setting up devices and using Change Tracker to perform RDF capacity planning:

1. Identify the device group that you want to monitor. If the group does not exist, you can create it using the `symdmg` command. For example, to create a device group named `deltadg`, enter:

```
symdmg create deltdag
```

2. Add a device (Symmetrix device name 085 on Symmetrix number 000000003264) to the device group:

```
symld -g deltdag -sid 3264 add dev 085
```

By default, SYMCLI assigns a logical device name (of the form DEVnnn) to this device.

3. Create a Change Tracker session for the device group:

```
symchg -g deltdag create
```

4. List the sessions that have just been created for the device group:

```
symchg -session list
```

5. Mark the device group (this is the starting point from which changes will be measured):

```
symchg -g deltdag mark
```

6. In a log file, record cumulative track changes that occur on each device in the device group. This example collects 60 samples at two-minute (120 seconds) intervals in a log file named `deltalog`:

```
symchg -g deltdag -log deltdalog -i 120 -c 60 view
```

7. Use the collected data from the log file for RDF capacity planning by providing two of the following three variables to a `symchg report` command:

- ◆The number of links
- ◆The link bandwidth
- ◆The amount of time available to synchronize the source (R1) and target (R2) devices

The `symchg report` command calculates the third variable. Refer to [Chapter 16, "Change Tracker Examples,"](#) for additional examples on performing `symchg report` calculations.

This chapter describes the function of gatekeepers and how to manage them in your system.

- ◆ Overview.....10-2
- ◆ SYMCLI gatekeeper management10-5

Overview

Low-level I/O commands executed using SYMCLI are routed to the Symmetrix array via a Symmetrix storage device that is specified as a *gatekeeper*. The gatekeeper device allows SYMCLI commands to retrieve configuration and status information from the Symmetrix array without interfering with normal Symmetrix operations. The gatekeeper must be accessible from the host where the commands are being executed.

Note: If the SYMCLI is running on a service processor, there is only one communication path (gatekeeper). Therefore, there is no need to manage it with `symgate`.

In addition to managing gatekeepers via the SYMCLI, gatekeepers management can also be handled automatically via the Base Daemon. For more information, refer to the *EMC Solutions Enabler Installation Guide*.

Gatekeeper candidates

All devices in your storage array are potential gatekeeper candidates. While you can explicitly dedicate devices as gatekeepers, and even associate those gatekeepers with specific device groups (as described later), SYMCLI can utilize any available device as a gatekeeper.

Note: BCV devices *cannot* be used or defined as gatekeepers. VCMDB devices can be used as a gatekeeper (but should not) but *cannot* be defined as a gatekeeper.

SYMCLI selects a gatekeeper based upon a pre-established priority list. A list of available gatekeepers can be displayed using the SYMCLI `symgate` command. The gatekeeper priority list includes all available physical device names prioritized from most likely gatekeeper to least likely, as shown below:

1. Defined and associated gatekeepers — These devices have been explicitly designated as gatekeeper devices and associated with specific device groups.
2. Defined gatekeepers — These devices have been explicitly designated as gatekeeper devices.
3. Small (≤ 10 cylinders) non-PowerPath devices, marked by the Symmetrix array with the inquiry gatekeeper flag.
4. Any small device in the following priority:
 - PowerPath parent device
 - PowerPath child device
 - PowerPath sibling
 - Count key data format
5. Standard non-RDF and non-meta devices in the following priority:
 - Non-PowerPath
 - PowerPath parent device
 - PowerPath child device
 - Count key data format
6. RDF R1 devices in the following priority:
 - Non-PowerPath
 - PowerPath parent device
 - PowerPath child device

7. RDF R2 devices in the following priority:

- Non-PowerPath
- PowerPath parent device
- PowerPath child device

How SYMCLI uses gatekeepers

When choosing a gatekeeper to process low-level I/O commands, SYMCLI will start with the highest priority candidate (Number 1), and try each possibility until successful.

For example, if you define a gatekeeper, but do not associate it to the device group (Number 2), SYMCLI chooses the first gatekeeper on the list. If that device is not accessible or is currently in use, SYMCLI will try using other gatekeeper devices in the gatekeeper list. If you do not define a gatekeeper, SYMCLI selects gatekeepers according to a dynamic selection algorithm (Numbers 3–7).

Note: It is recommended that you designate gatekeepers based upon your determined processing needs.

Once a gatekeeper has been successfully obtained, SYMCLI determines if a semaphore exists; if so, a lock is obtained on the device; otherwise, a semaphore is created and the device is then locked. Once the CDB sequence is processed, the gatekeeper is closed and the lock released, freeing the device for other processing.

Defined gatekeepers

A device can be defined as a gatekeeper device or not. Devices are defined using `symgate define`, which places them higher on the gatekeeper candidate priority list. Because devices that are defined as gatekeepers are likely to be used as a gatekeepers, it is recommended that these devices not be used (if possible) by the host system for normal data processing.

Note: Specifying a device as a gatekeeper that will also be used for normal data processing may result in a slight performance degradation on the given device. For this reason, it is not recommended.

A gatekeeper device must be visible to the local host.

How to size a gatekeeper

When a Symmetrix array is installed, the EMC Customer Engineer usually configures some Symmetrix devices with less than 10 cylinders (less than 5 MB) for use as gatekeeper devices.

Gatekeeper devices must be at least as large as the minimum volume size accessible by your host, which is usually 6 cylinders, or 2.8 MB. Consult your host documentation for the minimum device size accessible by your particular host to determine the minimum gatekeeper device size for your environment.

You can determine the storage size of a Symmetrix device using:

- ◆ The `sympd` command along with the `list` and `show` arguments as follows:
 - `list` — Displays a list of physical device names and storage size (in MBs) for a specific Symmetrix array.
 - `show` — Displays the parameters of a specified physical device that includes the device capacity or size in blocks and megabytes.
- ◆ The `syminq` command, specifying the physical device name.

Note: Although the EMC Customer Service Engineer usually configures a few Symmetrix devices for use as gatekeepers, these devices can be distinguished in a list executed by `syminq [PdevName]`, which shows a GK symbol next to the physical device name. Otherwise, they are not easily distinguished from other devices in other configuration lists, except perhaps by their size, which tends to be smaller than other devices.

Database considerations

Gatekeeper assignments reside in the Symmetrix host's configuration database. Whenever you add or remove a gatekeeper from the list or from a device group, the changes are updated to the database file (on disk).

SYMCLI gatekeeper management

Gatekeepers can either be managed automatically via the Base Daemon, or manually via the SYMCLI `symgate` command. This section focuses on managing gatekeepers via the SYMCLI. For information on managing gatekeepers via the Base Daemon, refer to the *EMC Solutions Enabler Installation Guide*.

You can use SYMCLI to prioritize the use of certain devices as gatekeepers by defining them as gatekeepers and associating them with specific device groups. SYMCLI maintains a list of these gatekeepers in the configuration database file. For a gatekeeper candidate priority list, refer to ["Gatekeeper candidates" on page 10-2](#).

Note: Choosing to use the Base Daemon to automatically manage gatekeepers will override any actions performed via `symgate`.

Defining a gatekeeper

Use `symgate` to define a device as a gatekeeper device. This command adds the specified physical device name to the list of gatekeeper devices.

For example on UNIX, to define device `c2t0d2` as a gatekeeper device, enter:

```
symgate define pd c2t0d2
```

On Windows, to define device `\\.\physicaldrive2` as a gatekeeper, enter:

```
symgate define pd physicaldrive2
```

SYMCLI uses one of the devices in the list as a gatekeeper, unless you have associated a gatekeeper explicitly with the device group of the device to which SYMCLI is sending a control command.

Note: You must add a device to the gatekeeper device list before you can associate it with a device group.

Undefine a gatekeeper

Use `symgate` to remove a Symmetrix device from the gatekeeper device list.

For example, to remove Symmetrix device `000C` from the gatekeeper device list, enter:

```
symgate undefine dev 000C
```

If the gatekeeper is associated with a device group, the gatekeeper must first be disassociated from the device group using `symgate` (also, refer to ["Disassociating from a device group" on page 10-6](#)).

Listing the gatekeeper devices

Use `symgate` to get a list of gatekeeper devices that have previously been defined as gatekeepers using `symgate define`.

For example, to get a list of gatekeeper devices belonging to Symmetrix 010010005104, enter:

```
symgate -sid 5104 list
```

To list all the physical device names in the gatekeeper list, enter:

```
symgate list
```

Associating with a device group

You can associate one or more gatekeepers with a device group. SYMCLI routes SCSI commands to any device in the device group via a gatekeeper. Use `symgate` to associate a gatekeeper with a device group.

For example, to associate a gatekeeper device `c2t0d2` with the device group `prod`, enter:

```
symgate -g prod associate pd c2t0d2
```

The gatekeeper device cannot be a member of any device group. You must add the gatekeeper to the gatekeeper list using `symgate define` before you can associate it with a device group. Use `symdg show` to list the gatekeepers associated with a device group.

Disassociating from a device group

Use `symgate` to disassociate a gatekeeper from a device group.

For example, to disassociate a gatekeeper device `c2t0d2` from a device group `prod`, enter:

```
symgate -g prod disassociate pd c2t0d2
```

This chapter describes how to configure, manage, and query Symmetrix events and logs.

- ◆ [Symmetrix event logs.....](#) 11-2

Symmetrix event logs

SYMCLI and SYMAPI normally log significant events and actions to a daily log file. On UNIX, the log file has the following pathname:

```
/var/symapi/log/symapi-yyyymmdd.log
```

On Windows, the log file has the following pathname:

```
C:\Program Files\EMC\Symapi\log\symapi-yyyymmdd.log
```

where:

yyyy — year

mm — month

dd — day

The log displays the following items concerning each event:

- ◆ Time tag of the event occurrence
- ◆ Process ID (PID)
- ◆ Source of the event (application name)
- ◆ Related (internal) API function call
- ◆ Name of the specific operation or event
- ◆ Variable event field that describes in detail the event or error

Note: Log files accumulate over time and can consume needed disk space. Periodically, you may need to purge the log files to conserve space.

For a detailed list of possible Symmetrix events, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

How to configure log options

The following options can be configured to customize how and where events are logged in your Symmetrix environment.

Table 11-1 Log file configuration options

Logging option	Configuration description
Enable/Disable Logging	You can disable logging by setting the environment variable SYMCLI_NOLOGGING to 1. For example, to disable logging on UNIX (C shell) enter: <pre>setenv SYMCLI_NOLOGGING 1</pre> To turn logging back on, enter: <pre>unsetenv SYMCLI_NOLOGGING</pre>
Log File Name	You can change the name and path of the default log file. For example, to change the log file name on UNIX (C shell), use the following form: <pre>setenv SYMCLI_LOG filename</pre> To turn daily log files back on, enter: <pre>unsetenv SYMCLI_LOG</pre>

Table 11-1 Log file configuration options (continued)

Logging option	Configuration description
Allow Undated SYMAPI Log File Names	You can allow the creation of undated SYMAPI log files by setting the environment variable SYMAPI_DATED_LOGFILE_NAME in the options file to <code>disable</code>. To allow undated log file names, set: <pre>SYMAPI_DATED_LOGFILE_NAME=DISABLE</pre> To re-enable dates, set: <pre>SYMAPI_DATED_LOGFILE_NAME=ENABLE</pre>
Change Date Formats	You can change date formats in the log entries by setting the environment variable SYMAPI_LOGFILE_DATE_FORMAT in the options file to <code>FORMAT2</code>. This formats the date as <code>yyyy-mm-dd</code>. To change the date format, set: <pre>SYMAPI_LOGFILE_DATE_FORMAT=FORMAT2</pre> To change the date format back to the default (<code>mm/dd/yyyy</code>), set: <pre>SYMAPI_LOGFILE_DATE_FORMAT=FORMAT1</pre>
Log File Configuration Options	You can change format of optional fields within each log record by setting the environment variable SYMAPI_LOGFILE_FORMAT in the options file. Zero or more of the following optional fields can be included: • <code>pid</code>—include the process ID. • <code>tid</code>—include the thread ID. • <code>userid</code>—include the user ID (useful with User Authorization is enabled). • <code>activityid</code>—include the activity ID. The default value is <code>pid tid</code>. To set the log file format, create an entry in the options file with value separated by spaces. For example: <pre>SYMAPI_LOGFILE_FORMAT = userid pid tid activityid</pre>
Log File Retention	Log File Retention is an option set in the options file to specify the number of days to retain the log files. • Maximum value: 1825 (or 5 years) • Minimum value: 6 • Alternative value: 0 (setting it to zero maintains the log file forever. This is the default for everything except the service processor whose default is 30.) For example, to change the Log File Retention to 60 days, enter: <pre>SYMAPI_LOGFILE_RETENTION=60</pre> Note: When the Log File Retention option is set, it overrides all existing Log File Retention values including those set for the service processor during initialization. The default for the service processor is 30 days. Setting this options file value will override that default. In addition, if you have accumulated many years of log files prior to setting this option, when the option is set it deletes those log files that do not meet the specified criteria.

Monitoring Symmetrix events via SYMCLI

The `symevent` command allows an administrator to monitor or track events within a Symmetrix array that may affect its operation. In some cases, reported events represent conditions that have already been repaired. This command allows the user to monitor the Symmetrix array for all reported events.

You can use `symevent` to either:

- ◆ **Monitor** — This action causes the command to run in the foreground, polling the Symmetrix array for new events every interval in seconds, either until the iteration count is satisfied or the program is stopped.
- ◆ **List** — This action examines the history of events, stored on the Symmetrix array, for those that meet the requested criteria. The `-start` and `-end` options allow the user to retrieve events that occurred between certain time bounds. In addition, you can sort the returned events based upon the reporting director using the `-DIR` option.

You can further refine your query to a Symmetrix ID and restrict the reported events to those of a minimum severity (warnings, errors, or fatal events).

Monitor

In its most basic invocation, `symevent` polls for and reports on all events, on all locally-connected Symmetrix systems, every 10 seconds, forever:

```
symevent monitor
```

The following example polls for and displays events of severity `Warning` or greater on Symmetrix 012345678901 every 10 minutes for a 24-hour period:

```
symevent monitor -sid 012345678901 -i 600 -c 144 -warn
```

Example output from the previous command:

```
Symmetrix ID: 012345678901
```

Detection time	Src	Category	Severity	Error Num
Wed Dec 15 09:00:40	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 10:01:57	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 11:02:29	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 12:02:29	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 13:03:42	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 14:04:13	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 16:04:13	2006 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
. . .				

- List** The following example illustrates how to retrieve a verbose list of the events which have occurred on the specified Symmetrix array between 9am and 5pm today:

```
symevent list -sid 78901 -v -start 09:00 -end 17:00
```

The following is an example output from the previous command:

Symmetrix ID: 012345678901

Event at Wed Dec 15 12:48:26 2006:

```
Reporting Director : RF-16A
Source             : Symmetrix
Category           : RDF
Severity           : Informational
Numeric Code       : 82.0467.00
Description        : The RDF subsystem initiated a SIM message to a
Symmetrix remotely-attached via RDF links
```

Event at Wed Dec 15 12:48:49 2006:

```
Reporting Director : RF-16A
Source             : Symmetrix
Category           : RDF
Severity           : Informational
Numeric Code       : 82.0467.00
Description        : The RDF subsystem initiated a SIM message to a
Symmetrix remotely-attached via RDF links
```

. . .

Common audit log

Data is written to a common audit file during Symmetrix control operations initiated by host applications. The common audit log correlates activity from all hosts into one file that is stored in the Symmetrix File System (SFS).

The `symaudit` command enables the examination and filtering of the common audit log file for a specified Symmetrix array. The audit log resides on the Symmetrix array and currently has a maximum size of 40 MB. Once the 40 MB limit is reached, the log begins to overwrite itself. You don't have to do anything to maintain this file, unless you want to capture records before the circular 40 MB space recycles.

Use the following actions to parse the audit log contents:

- ◆ `show` — Displays details about the audit log itself for a specific Symmetrix array, including: the total range of records, the date/time range, and the starting record number.
- ◆ `list` — Lists details about the requested records in either a brief or verbose format. The range of records to extract can be filtered by one or all of the following: record number, record count, date/time, functional area, control action, vendor ID, application ID, host name, user name, and device acted upon.

Note: For details on valid parameter values on which to filter, refer to the `symaudit` command in the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Example output

The following is an example output of the `show` action for a specified Symmetrix array:

```
symaudit -sid 04 show

                A U D I T   L O G   D A T A

Symmetrix ID           : 000000006204

Starting date           : 05/26/2006 12:55:39
Ending date             : 07/12/2006 10:22:27

Starting record number  : 175500
Ending record number    : 237298
Total record count      : 61799
```

The following is an example output of the `list` option for a specified Symmetrix array, beginning at record number 44 and displaying 5 records:

```
symaudit -sid 04 list -v -start_time 7/12:10:10 -end_time 7/12:10:20

                A U D I T   L O G   D A T A

Symmetrix ID           : 000000006204

Record Number          : 237291
Records in Seq         : 1
Offset in Seq          : 1
Time                   : 07/12/06 10:19:47
Vendor ID              : EMC Corp
Application ID          : SYMAUTH
Application Version     : 6.3.0.106
API Library            : SDK
API Version            : X6.3.0.10 (Edit Level: 762)
Host Name              : api196
OS Name                : SunOS
OS Revision            : 5.8Generic
Client Host            :
Process ID             : 00029742
```

```

Task ID           : 00000001
Function Class    : ACCESS
Action Code       : Add
Text              : Starting a User Authorization
                   operation to set role for user
                   paulr to Monitor
Username          : H:api196\ruggip
Activity ID       : SEa5e5850490

Record Number     : 237292
Records in Seq    : 1
Offset in Seq     : 1
Time              : 07/12/06 10:19:47
Vendor ID         : EMC Corp
Application ID     : SYMAUTH
Application Version : 6.3.0.106
API Library       : SDK
API Version       : X6.3.0.10 (Edit Level: 762)
Host Name         : api196
OS Name           : SunOS
OS Revision       : 5.8Generic
Client Host       :
Process ID        : 00029742
Task ID           : 00000001
Function Class    : ACCESS
Action Code       : Add
Text              : The User Authorization set
                   role operation SUCCEEDED
Username          : H:api196\ruggip
Activity ID       : SEa5e5850490

```

- ◆ Record Number — The current record number.
- ◆ Records in Seq — Total number of records requested.
- ◆ Offset in Seq — Offset number from the first record requested.
- ◆ Time — Date and time the record was entered.
- ◆ Vendor ID — ID of the vendor whose application logged the record.
- ◆ Application ID — ID of the application that logged the record.
- ◆ Application Version — Application version number.
- ◆ API Library — Name of the SYMAPI library the application ran against.
- ◆ API Version — Version of the SYMAPI.
- ◆ Host Name — Name of the host that logged the record.
- ◆ OS Name — Operating system on which the host is running.
- ◆ OS Revision — Operating system revision number.
- ◆ Client Host — Any SYMCLI client communicating with the SYMAPI server.
- ◆ Process ID — ID of the process that logged the record.
- ◆ Task ID — ID of the task.
- ◆ Function Class — Class name of the SYMAPI functional area.
- ◆ Action Code — Name of the SYMAPI control action associated with an audit log entry.
- ◆ Text — Text details of the given entry.
- ◆ Username — Identifies the user that generated the log entry.
- ◆ Activity ID — A randomly generated ID that uniquely identifies this action.

Event daemon and SNMP traps

Solutions Enabler also provides an asynchronous event daemon that can be used to monitor and manage Symmetrix events. The event daemon (`storevntd`) provides the necessary MIB maintenance and SNMP trap generation services required to monitor the status of Symmetrix storage environments from third-party enterprise management frameworks.

To learn more about enabling and configuring the event daemon, refer to the *EMC Solutions Enabler Installation Guide*.

This chapter describes the XML output option of the SYMCLI.

- ◆ [XML structured output data](#)12-2

XML structured output data

The XML (*Extensible Markup Language*) output option provides a mechanism to facilitate the automated processing of SYMCLI output data. XML is a deterministic parsing tool that eases parsing of output data, providing a functional advantage over screen-scraping tools like awk or Perl. The XML industry standard is based on the experience of SGML and is endorsed by the World Wide Web Consortium. Detailed information on XML may be found at:

<http://w3.org/XML/>

XML has the look and feel of HTML, as it employs the same tag-based syntax. However, XML uses tags to delimit data—as opposed to defining the data as with HTML—allowing the document author to specify the tags most applicable to the given application. For the SYMCLI, tags represent the physical and logical structures within the Symmetrix array and its environments.

When XML mode is utilized, the data returned is identical to that of the *standard* output, but “marked-up” with tags. These tags enable individual pieces of data to be readily called upon by name. In addition, they provide a definitive way to express the relationship between different objects; an advantage over the standard CLI display output.

XSLT: XML data transformations

Many tools are available to query, filter, retrieve, and format specific information stored in complex XML files. Among these, eXtensible Stylesheet Language Transforms (XSLT) is a particularly useful and widely available technology. While using XML will result in less ambiguous, more robust scripts; XSLT will make the information presented in XML accessible to the more familiar plain text-based scripting techniques.

To introduce you to XSLT, a directory containing several examples of the types of queries that can be performed on XML data using XSLT is provided. The examples are designed only to provide a brief introduction to the power and usefulness of XSLT, and can help ease the transition to XML.

XML output options

In XML, data can be described either by using elements or attributes, or the combination of both. Versions of Solutions Enabler prior to version 6.3 only provided attribute-based XML data. Beginning in version 6.3, you can now return element-based XML output.

Element-based XML describes data in a hierarchical manner by using the notion of parent and children. An element can have several different content types. It can have element content (child element), a mixed content which contains both text and child element, a simple content containing text only, or an empty content carrying no information. An element can also have attributes. These additional content types would allow users to modify the data structures in a fairly flexible manner.

On the other hand, an attribute is used to provide additional information about an element. An attribute is in general used to store the metadata describing the data that stored in XML. Although data can be stored in attributes, it is best practice to store data in child elements.

Note: The attribute-based XML output (`-output xml`) has not been updated with Solutions Enabler version 6.3 features and is maintained strictly for backwards compatibility. In order to maintain the consistency between an XML output and a standard output, new elements will be added to the element-based XML output.

Using XML Mode

To use XML mode with SYMCLI, an environment variable or a command line option can be used.

Use the environment variable to globally set your command output to XML or globally return it to standard mode. Use the following syntax to set the environment variable:

```
SYMCLI_OUTPUT_MODE = xml|xml_element|standard
```

- ◆ **xml** — returns the output of all legacy commands in attribute-based XML tags.
- ◆ **xml_element** — returns the output of all commands in element-based XML tags.
- ◆ **standard** — returns the output of all commands to the default output without the XML tags.

Note: When the environment variable output mode is set to `xml` or `xml_element`, commands that do not support XML output will generate a runtime error message. You can override this behavior by setting the command line `-output` option to a value of `standard`, as described below. This will allow you to successfully execute the given command in standard mode.

The command line method provides a means of outputting any single command as either XML or standard output. Using the command line method overrides the current environment variable setting:

```
<SymcliCommand> -output <xml|xml_element|standard>
```

The `-output` flag is not found in `-help` or `man` pages because of its wide scope and usage.

Example Output

The following examples show some of the differences between the attribute- and element-based XML output.

Attribute-based XML

```
<Symmetrix id="0000000000001", attachment="local">
  <Device dev_name="0001"/>
</Symmetrix>
```

Element-based XML

Notice that a new element tag `<Symm_Info>` has been added to the XML data shown below to store the general Symmetrix data.

```
<Symmetrix>
  <Symm_Info>
    <id>0000000000001</id>
    <attachment>local</attachment>
  </Symm_Info>
  <Device dev_name="0001"/>
</Symmetrix>
```

Note: In order to maintain consistent element names in all SYMCLI commands, some tag names will be redefined. For example, in current SYMCLI command, different names exist to describe Symmetrix identification number, such as `id`, `symmetrix`, or `symid`. A new consistent tag name will be defined across all SYMCLI commands in the element-based XML output.

The Operational Examples part of this product guide identifies and focuses on some specific array management operations most typically used in the maintenance of your Symmetrix storage environment. These practical examples illustrate various array control processes by showing the SYMCLI command sequences to accomplish these tasks. These specific management tasks are described in the subsequent chapters as follows:

[Chapter 13, “Symmetrix Configuration Information Examples,”](#) provides realworld examples of how to obtain various Symmetrix configuration information.

[Chapter 14, “Access Control Examples,”](#) provides real-world examples of using Symmetrix Access Control.

[Chapter 15, “GNS Examples,”](#) provides realworld examples of how to use the optional GNS functionality.

[Chapter 16, “Change Tracker Examples,”](#) provides examples of the SYMCLI actions and specific commands, which are used to measure and track changes to data on a group of Symmetrix volumes.

This chapter provides real world examples of how to obtain various Symmetrix configuration information.

- ◆ Example 1: Verifying SYMAPI database configuration information 13-2
- ◆ Example 2: Obtaining more Symmetrix information 13-4
- ◆ Example 3: Obtaining more device information 13-7
- ◆ Example 4: Obtaining information about dynamic RDF devices 13-11
- ◆ Example 5: Obtaining information about virtual devices 13-12
- ◆ Example 6: Obtaining more device group information 13-15
- ◆ Example 7: Obtaining disk and audit information 13-21
- ◆ Example 8: Obtaining display output in XML format 13-26
- ◆ Example 9: Obtaining status information about GNS 13-28

Example 1: Verifying SYMAPI database configuration information

Whenever a Symmetrix configuration is new or has changed, you need to ensure which Symmetrix arrays are connected to your host and that the information about these Symmetrix arrays is current.

The `symcfg discover` command builds or rebuilds your host's SYMAPI database file with the most current information about the physical devices directly connected to your host. The command scans all buses, collects information about all the devices found, and rebuilds the database with collected information from all locally and remotely attached Symmetrix devices.

symcfg discover

This operation may take up to a few minutes. Please be patient...

The `symcfg list` command displays the configuration information from the local host's SYMAPI database file.

symcfg list

```

                                S Y M M E T R I X

SymmID      Attachment  Model      Mcode      Cache      Num Phys      Num Symm
              Version    Size (MB)   Devices    Devices
000185500814 Local        DMX800     5670        6144        204         938
000185500505 Remote       DMX800     5670        6144         0         763

```

The `symcfg verify` command determines if the Symmetrix configuration and the SYMAPI database file are in sync. If they are out of sync, the verify action returns code 24 (CLI_C_NOT_IN_SYNC). If they are in sync, the verify action returns code zero (the CLI_C_SUCCESS value) as shown below using the `echo $status` command.

symcfg verify -sid 814

The Symmetrix configuration and the database file are in sync.

```
echo $status
0
```

A new device is now added to local Symmetrix 814, changing the configuration. The `verify` action is repeated, and its output shows an out-of-sync condition. The `echo $status` command returns the value 24, confirming that the configuration and the database configuration information are now out of sync.

symcfg verify -sid 814

The Symmetrix configuration and the database file are NOT in sync.

```
echo $status
24
```

The `symcfg discover` command rebuilds your host's SYMAPI database file with the most current information.

symcfg discover

This operation may take up to a few minutes. Please be patient...

The `symcfg list` command now displays the updated configuration information from the local host's rebuilt SYMAPI database file. Note that the number of Symmetrix devices displayed for Symmetrix 814 has increased from 938 (shown in the previous `symcfg list` command) to 939.

`symcfg list`

```

                                S Y M M E T R I X

SymmID      Attachment  Model      Mcode    Cache    Num Phys  Num Symm
              Version    Size (MB)  Devices  Devices
000185500814 Local        DMX800    5670      6144     204      939
000185500505 Remote        DMX800    5670      6144      0      763

```

The `symcfg verify` command confirms that the configuration and SYMAPI database file are now in sync again.

`symcfg verify -sid 814`

The Symmetrix configuration and the database file are in sync.

Example 2: Obtaining more Symmetrix information

This example illustrates some of the more useful `symcfg list` options.

Using the `-connections` option with `symcfg list` command allows you to see the host connections to a Symmetrix array. Only those hosts that have at least one registered application are listed. The node name `hk185500814` is the service processor.

`symcfg list -connections`

Symmetrix ID : 000185500814

Symmetrix		Host				
Director	Port	Node Name	IP Address	HW Type	OS Name	OS Revision
DA-1A	0	hk185500814	172.26.131.103	INTEL	WinNT	4.0.1381
SA-4A	0	api183	172.23.65.183	9000/800	HPUX	B.11.00
SA-13A	0	api183	172.23.65.183	9000/800	HPUX	B.11.00
SA-4B	0	api212	172.23.65.212	9000/800	HPUX	B.11.00
FA-5B	0	api179	172.23.65.179	sun4u	SunOS	5.8
FA-12B	0	api179	172.23.65.179	sun4u	SunOS	5.8

Using the `-sorthost` option with the `-connections` option and `symcfg list` command allows you to display the host connections to Symmetrix arrays by host rather than by Symmetrix ID. The ellipsis (.....) represents truncated output.

`symcfg list -connections -sorthost`

Host				Symmetrix		
Node Name	IP Address	OS Name	OS Revision	ID	Director	Port
hk185500814	172.26.131.103	WinNT	4.0.1381	000185500814	DA-1A	0
api179	172.23.65.179	SunOS	5.8	000185500814	FA-5B	0
				000185500814	FA-12B	0

External locks are used to lock access to the entire Symmetrix array during critical operations. SYMAPI uses lock numbers 0 to 15; applications assigned by EMC use lock numbers greater than 15. By default, the `symcfg list -lock` command displays what Symmetrix arrays have a lock number of 0 (used to lock a Symmetrix during SRDF operations). The display shows two unlocked Symmetrix arrays.

`symcfg list -lock`

S Y M M E T R I X		L O C K S		
SymmID	Attachment	Lock Status	Lock Number	Time Held (Sec)
000185500814	Local	Unlocked	N/A	N/A
000185500505	Remote	Unlocked	N/A	N/A

Using the `-lockn` option with `symcfg list` displays Symmetrix arrays that have a specific lock number. Specifying zero (0) in the following command displays Symmetrix arrays that currently have an external lock for SRDF operations. That is, an SRDF control operation was executing when this command was issued. Using the `-lockn RDF` option displays Symmetrix arrays that have any of the sixteen locks (0x100 to 0x10F) used during SRDF operations. Using the `-lockn ALL` option displays these operation locks plus other locks used by such applications as Optimizer, Device Masking, and Configuration Manager.

```
symcfg list -lockn 0
```

S Y M M E T R I X		L O C K S		
SymmID	Attachment	Lock Status	Lock Number	Time Held (Sec)
000185500814	Local	Locked	0	8
000185500505	Remote	Locked	0	8

Using the `-applications` option with `symcfg list` displays all applications that are registered in a Symmetrix array and that have accessed that Symmetrix array. The following display lists such applications on a Symmetrix array (sid 814) connected to the local host.

```
symcfg list -applications
```

```
Symmetrix ID      : 000185500814
```

Host		Application			
Node Name	IP Address	ID	Vendor ID	Version	Attr
api179	172.23.65.179	SYMCLI	EMC Corp	5.0.0.1	-
		SYMCLI	EMC Corp	5.0.0.0	-
		EMCpower	EMC Corp	2.1.0.0	-
		SYMCLI	EMC Corp	5.0.0.32	-
		SYMCLI	EMC Corp	5.0.0.29	-
api180	172.23.65.180	SYMCLI	EMC Corp	5.0.0.32	(R)
hk185500814	172.26.131.103	OPTIMIZER	EMC Corp	2.0.0.4	-

Legend for Attribute(s):

(M) : Application Registered Remotely via RDF links (Multi-Hop).

(R) : Application Registered Remotely via RDF links (One-Hop).

Using the `-semaphores` option with `symcfg list` displays gatekeeper (GK), database (DB), and lock file (FILE) semaphores.

symcfg list -semaphores

```

                S Y M A P I      S E M A P H O R E S

ID      Key      State      Lock Proc.
-----
      1 0x45000327 Unlocked GK      0 /dev/rdisk/c4t0d103s2
      3 0x45000ba7 Unlocked GK      0 /dev/rdisk/emcpower218c
      1 0x45000327 Unlocked DB      0 /var/symapi/db/symapi_db.bin
      0 0x45000246 Unlocked FILE    0 /var/symapi/config/symapislck

```

Using the `-services` option with `symcfg list` command displays configured network services for use by the SYMAPI client. These services are listed in the network services file (`netcnfg`) in the `/var/symapi/config` directory.

symcfg list -services

```

                S Y M A P I      N E T      S E R V I C E S

Name      Domain Type      Node Name      Address      Port
-----
api182            TCPIP      api182            172.23.65.182      2707

```

Example 3: Obtaining more device information

This example illustrates some of the more useful `symdev list` options.

Device external locks

SYMAPI uses device external locks in a Symmetrix array to lock device pairs during control operations with TimeFinder family and SRDF family products. The following command displays external locks on standard devices on Symmetrix 814 during a SRDF/AR operation. That is, the command was issued in the middle of an SRDF/AR cycle.

```
symdev list -lock
```

Symmetrix ID: 000185500814

S Y M M E T R I X D E V I C E L O C K S						
Device Name			Device			
Sym	Physical	Lock Number	Lock Holder ID	Seconds Held	Lock Flags	Device Attribute
03F7	/dev/vx/rdmp/c4t0d107s2	9	0x10010FE6	111	Long	Grp'd
03F8	/dev/vx/rdmp/c4t0d108s2	9	0x10010FE6	111	Long	Grp'd
03F9	/dev/vx/rdmp/c3t4d3s2	9	0x10010FE6	111	Long	Grp'd
03FA	/dev/vx/rdmp/c3t0d0s2	9	0x10010FE6	111	Long	Grp'd

Multiple front-end SCSI adapter ports

Using the `-multiport` option with the `symdev list` command displays devices that are mapped to multiple front-end SCSI adapter (SA) ports. For example, the following display shows Symmetrix device 0000 mapped to SA ports 12B:0 and 05B:0. Other devices (Sym column) are mapped to these same ports.

```
symdev list -multiport
```

Symmetrix ID: 000185500814

M U L T I - P O R T				D E V I C E S			
Device Name		Directors		Device			
		Sym	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)
		0000	16B:D1	2-Way Mir	Grp'd	RW	4315
/dev/vx/rdmp/c4t0d0s2		- 12B:0	- -	-	-	-	-
/dev/vx/rdmp/c5t0d0s2		- 12B:0	- -	-	-	-	-
/dev/rdsk/emcpower116c		- 05B:0	- -	-	-	-	-
/dev/rdsk/c4t0d0s2		- 05B:0	- -	-	-	-	-
/dev/rdsk/c5t0d0s2		- 12B:0	- -	-	-	-	-
		0001	02B:D1	2-Way Mir	Grp'd	RW	4315
/dev/vx/rdmp/c4t0d1s2		- 12B:0	- -	-	-	-	-
/dev/vx/rdmp/c5t0d1s2		- 12B:0	- -	-	-	-	-
/dev/rdsk/emcpower117c		- 05B:0	- -	-	-	-	-
/dev/rdsk/c4t0d1s2		- 05B:0	- -	-	-	-	-
/dev/rdsk/c5t0d1s2		- 12B:0	- -	-	-	-	-
		0002	01B:D1	2-Way Mir	Grp'd	RW	4315
/dev/vx/rdmp/c4t0d2s2		- 12B:0	- -	-	-	-	-
/dev/vx/rdmp/c5t0d2s2		- 12B:0	- -	-	-	-	-
/dev/rdsk/emcpower118c		- 05B:0	- -	-	-	-	-
/dev/rdsk/c4t0d2s2		- 05B:0	- -	-	-	-	-
/dev/rdsk/c5t0d2s2		- 12B:0	- -	-	-	-	-

```

                                00C8          01A:C2 RDF2+Mir      N/Grp'd      WD      4315
Not Visible                    - 05A:0      - -              -              -              -
Not Visible                    - 12A:0      - -              -              -              -

                                00C9          15A:C2 RDF2+Mir      N/Grp'd      WD      4315
Not Visible                    - 05A:0      - -              -              -              -
Not Visible                    - 12A:0      - -              -              -              -

```

Mult-port: First mapped port

Using the `-firstport` option with the `symdev list` command displays just the first port of information for devices that are mapped to multiple ports.

```
symdev list -firstport
```

```
Symmetrix ID: 000185500814
```

	Device Name	Directors	Device			
Sym	Physical	SA :P DA :IT Config	Attribute	Sts	Cap (MB)	
0000	/dev/vx/rdmp/c4t0d0s2	12B:0 16B:D1 2-Way Mir	Grp'd	RW	4315	
0001	/dev/vx/rdmp/c4t0d1s2	12B:0 02B:D1 2-Way Mir	Grp'd	RW	4315	
0002	/dev/vx/rdmp/c4t0d2s2	12B:0 01B:D1 2-Way Mir	Grp'd	RW	4315	
0003	/dev/vx/rdmp/c4t0d3s2	12B:0 16A:D1 2-Way Mir	Grp'd	RW	4315	
0004	/dev/vx/rdmp/c4t0d4s2	12B:0 15A:D1 2-Way Mir	Grp'd	RW	4315	
0005	/dev/vx/rdmp/c4t0d5s2	12B:0 01A:D1 2-Way Mir	Grp'd	RW	4315	

SCSI reservations

Using the `-resv` option with the `sympd list` command displays all Symmetrix devices that have SCSI reservations.

```
sympd list -resv
```

```
Symmetrix ID: 000000003143
```

Device Name	Device	Reservation
Physical	Sym Config	Initiator SA :P
/dev/rhdisk20	0018 2-Way Mir	07 01C:0

Meta head members The `symdev list` command with the `-meta` option shows the meta head member on the meta device. *Concated* means that the meta device consists of a certain number (# of Devs heading in the output table) of concatenated devices. *Striped* means that the meta device is striped in Stripe Size chunks across # of Devs devices. Cap MB is the total capacity of the entire meta device (meta head plus meta members).

symdev list -meta

Symmetrix ID: 000185500814

Device Name		Device		Meta Information			
Sym	Physical	Config	Attr	Config	Stripe Size	# of Devs	Cap (MB)
0268	Not Visible	RDF2+Mir	N/Grp'd	Concated	-	2	17263
026A	Not Visible	RDF2+Mir	N/Grp'd	Striped	960k	3	25895
026D	Not Visible	RDF2+Mir	N/Grp'd	Striped	3840k	3	25895
0270	Not Visible	BCV	N/Asst'd	Concated	-	2	17263
0272	Not Visible	BCV	N/Asst'd	Striped	960k	3	25895
0275	Not Visible	BCV	N/Asst'd	Striped	3840k	3	25895
03C7	Not Visible	Unprotected	N/Grp'd	Striped	1920k	4	4125
03C8	Not Visible	Unprotected	N/Grp'd	Striped	7680k	4	4125
03D3	Not Visible	Unprotected	N/Grp'd	Concated	-	4	4125
03D6	Not Visible	Unprotected	N/Grp'd	Striped	30720k	4	4125
03D7	Not Visible	BCV	N/Asst'd	Striped	1920k	4	4125
03D8	Not Visible	BCV	N/Asst'd	Concated	-	4	4125
03DF	Not Visible	BCV	N/Asst'd	Striped	7680k	4	4125
03E0	Not Visible	BCV	N/Asst'd	Striped	1920k	4	4125
03E2	Not Visible	BCV	N/Asst'd	Striped	30720k	4	4125
03E5	Not Visible	BCV	N/Asst'd	Striped	7680k	4	4125
03F4	Not Visible	BCV	N/Asst'd	Concated	-	4	4125
03F5	Not Visible	BCV	N/Asst'd	Striped	30720k	4	4125
03FB	/dev/rdisk/emcpower229c	2-Way Mir	Grp'd	Striped	7680k	4	4125
03FF	/dev/vx/rdmp/c3t0d2s2	2-Way Mir	Grp'd	Concated	-	4	4125
0403	/dev/vx/rdmp/c3t0d3s2	2-Way Mir	Grp'd	Striped	15360k	4	4125
040B	Not Visible	RDF1-BCV+Mir	Asst'd	Striped	7680k	4	4125
040F	Not Visible	RDF1-BCV+Mir	Asst'd	Concated	-	4	4125
0413	Not Visible	RDF1-BCV+Mir	Asst'd	Striped	15360k	4	4125

Hotspares When a disk fails, a hot spare (a spare disk) is invoked against it. All devices that have mirrors (hypervolumes) on the failed disk against which the hot spare was invoked, are shown in the `symdev list -hotspare` display as having a hot spare invoked against them. This display shows that Symmetrix 120 has not had any hot spares invoked against failed disks, whereas Symmetrix 180 shows one hot (dynamic) spare invocation against disk 01A:D0. Note that Unprotected devices with a hot spare invoked against them become not ready (NR) because unprotected devices have no mirror to kick in to mirror the spare hypervolume. On the other hand, mirrored devices stay ready (RW) when a hot spare is invoked against them.

`symdev list -hotspare`

Symmetrix ID: 000184500120

Could not select any Symmetrix devices to list.

Symmetrix ID: 000184500180

Device Name		Directors			Device		
Sym	Physical	SA	:P DA	:IT Config	Attribute	Sts	Cap (MB)
0010	Not Visible	???:?	01A:D0	Unprotected	N/Grp'd	NR	103
0098	Not Visible	???:?	01A:D0	Unprotected	N/Grp'd	NR	47
00AE	Not Visible	13A:0	01A:D0	2-Way Mir	N/Grp'd	RW	4315
00C7	Not Visible	???:?	01A:D0	Unprotected	N/Grp'd	NR	47
00DF	Not Visible	???:?	01A:D0	Unprotected	N/Grp'd	NR	47
00F5	Not Visible	???:?	01A:D0	Unprotected	N/Grp'd	NR	47
0108	Not Visible	???:?	01A:D0	2-Way Mir	N/Grp'd	RW	469

Example 4: Obtaining information about dynamic RDF devices

This example looks for devices on a local source Symmetrix array (sid 814) that are capable of being dynamic RDF devices. Devices configured for RDF capability can be used to create dynamic SRDF pairs while the Symmetrix arrays are in operation (“on the fly”).

The `symcfg list` command displays high-level information on Symmetrix arrays in this configuration.

symcfg list

S Y M M E T R I X						
SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000185500814	Local	DMX800	5670	6144	204	938
000185500505	Remote	DMX800	5670	6144	0	763

With Engenuity version 5568 and higher, the `symdev list` command with the `-dynamic` option displays devices configured for dynamic RDF capability. This command displays devices that have been created as, or are capable of being created as, dynamic SRDF pairs using the `symrdf createpair` command. When combined with the `-r1` option, `symdev list -dynamic` displays devices configured for dynamic R1/R2¹ and R1-only; when combined with the `-r2` option, the command displays devices configured for dynamic R1/R2 and R2-only. RDF1+Mir in the display indicates devices that have already been created as dynamic RDF devices.

symdev list -sid 814 -dynamic -r1

Symmetrix ID: 000185500814

Device Name		Directors		Device			
Sym	Physical	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)
00F2	/dev/rdisk/emcpower224c	04A:0	15B:C4	Unprotected	N/Grp'd	RW	2063
00F5	/dev/rdisk/emcpower227c	04A:0	01A:D1	Unprotected	N/Grp'd	RW	2063
00FA	/dev/rdisk/emcpower239c	04A:0	15A:C4	Unprotected	N/Grp'd	RW	2063
0106	/dev/rdisk/emcpower235c	04A:0	01B:D3	Unprotected	N/Grp'd	RW	2063
010A	/dev/rdisk/emcpower237c	04A:0	15A:D3	Unprotected	N/Grp'd	RW	2063
010F	/dev/rdisk/emcpower240c	04A:0	02B:C2	Unprotected	N/Grp'd	RW	2063
0145	Not Visible	04A:0	01A:C1	RDF1+Mir	N/Grp'd	RW	2063
0146	Not Visible	04A:0	15A:C1	RDF1+Mir	N/Grp'd	RW	2063
0147	Not Visible	04A:0	02A:C1	RDF1+Mir	N/Grp'd	RW	2063
0148	Not Visible	04A:0	15A:D1	RDF1+Mir	N/Grp'd	RW	2063
0149	Not Visible	04A:0	02A:D1	3-Way Mir	N/Grp'd	RW	2063
014A	Not Visible	04A:0	16B:C4	3-Way Mir	N/Grp'd	RW	2063
014B	Not Visible	04A:0	02B:C4	3-Way Mir	N/Grp'd	RW	2063

1. Devices intended for dynamic RDF swap must be configured with the `dyn_rdf` attribute, which allows a device to be either an R1 or an R2 device.

Example 5: Obtaining information about virtual devices

The virtual device is a host-addressable Symmetrix device with special attributes created at configuration time. But unlike the BCV, which contains a full volume of data, the virtual device is a “slim” device that offers a space-saving way to create instant point-in-time copies of logical volumes. Any updates to a source device after its activation with a virtual device cause the pre-update image of the changed tracks to be copied to a *save device*. The virtual device’s indirect pointer is then updated to point to this original track data, preserving a point-in-time image of the volume. TimeFinder/Snap uses this copy-on-first-write technique to conserve disk space, since only changes to tracks on the source cause any incremental storage to be consumed. You use `symsnap` commands to create and activate a source/target *snap pair*.

This example looks on the local Symmetrix array (000000006206) for devices configured as virtual devices.

The `symdev list` command displays devices on the Symmetrix array that are configured as target virtual devices (VDEV). Devices with the SV attribute have been configured as save devices. To list VDEVs only, use `symdev list -vdev`. The ellipsis (.....) indicates where superfluous output was omitted or truncated.

symdev list

Symmetrix ID: 000000006206

Device Name		Directors		Device		
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)
0000	Not Visible	???:? 01A:C0	BCV	Asst'd	RW	8632
0001	Not Visible	???:? 16C:D0	BCV	Asst'd	RW	8632
0002	Not Visible	???:? 01B:D0	BCV	Asst'd	RW	8632
0003	Not Visible	???:? 16D:C0	BCV	Asst'd	RW	8632
0048	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	RW	8632
0049	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	RW	8632
004A	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	RW	8632
004B	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
004C	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
004D	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
004E	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
004F	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
0050	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
0051	Not Visible	???:? ??:?:??	VDEV	N/Grp'd	NR	8632
0052	Not Visible	???:? 16B:D1	2-Way Mir	N/A (SV)	RW	8632
0053	Not Visible	???:? 01C:C0	2-Way Mir	N/A (SV)	RW	8632
0054	Not Visible	???:? 16B:C0	2-Way Mir	N/A (SV)	RW	8632

The `symlld list` command displays devices that have been added to a device group named 3vdevs. The source of any snap operation is standard device 0076. The targets can be three virtual devices (004F, 0050, and 0051) that are paired with the standard device. At different points in time, each virtual device can be the target of snap operation. The virtual devices do not yet have copy sessions with the standard device (indicated by their status of NotCreated).

symlld list

```
Device Group (DG) Name: 3vdevs
DG's Type           : REGULAR
DG's Symmetrix ID    : 000000006206
```

Standard Device Name			Directors		Device		
Logical	Physical	Sym	SA :P DA :IT	Config	Att Sts	Cap (MB)	
DEV001	DRIVE29	0076	04A:0 16C:D2	2-Way Mir	RW	8632	

Legend for STD devices:

```
(+): Paired with a BCV device that is associated with this dg.
(-): Paired with a BCV device that is non-associated with this dg.
```

Symmetrix Virtual Devices associated with this dg:

VDEV Device				Source Device		Status
Logical	Sym	Att.	Sts	Logical	Sym	SRC <=> VDEV
VDEV001	004F	NR	N/A		N/A	NotCreated
VDEV002	0050	NR	N/A		N/A	NotCreated
VDEV003	0051	NR	N/A		N/A	NotCreated

Legend for VDEV devices:

```
(+): VDEV is paired with a member device.
(-): VDEV is paired with a non-member device.
```

The `symsnap list` command indicates whether there are snap copy sessions running on the Symmetrix array. This display shows that there are no copy sessions currently on Symmetrix 6206.

symsnap list

```
Symmetrix ID: 000000006206
```

```
No Snap sessions found
```

The `symsnap list` command with the `-savedevs` option displays any save devices that have been configured to hold data for virtual devices. This display indicates a list of save devices and that there has not yet been any virtual device activity that caused tracks of data to be copied to these save devices.

symsnap list -savedevs

Symmetrix ID: 000000006206

Snap Save Devices					
Sym	Device Type	Total Tracks	Used Tracks	Free Tracks	Percent Full
0052	FBA	276210	0	276210	0%
0053	FBA	276210	0	276210	0%
0054	FBA	276210	0	276210	0%
0055	FBA	276210	0	276210	0%
0056	FBA	276210	0	276210	0%
0057	FBA	276210	0	276210	0%
0058	FBA	276210	0	276210	0%
0059	FBA	276210	0	276210	0%
005A	FBA	276210	0	276210	0%
005B	FBA	276210	0	276210	0%
Total					
Tracks		2762100	0	2762100	0%
MB(s)		86315.6	0.0	86315.6	

The `symsnap query` command displays three snap copy sessions that are controlled within the device group named `3vdevs`. The `-multi` option is required to display multiple sessions.

symsnap query -g 3vdevs -multi

Device Group (DG) Name: 3vdevs
 DG's Type : REGULAR
 DG's Symmetrix ID : 000000006206

Source Device			Target Device			State		Copy
Logical	Sym	Protected Tracks	Logical	Sym	G	Changed Tracks	SRC <=> TGT	(%)
DEV001	0076	276210	VDEV003	0051	X	0	CopyOnWrite	0
		273902	VDEV002	0050	X	2308	CopyOnWrite	0
		273603	VDEV001	004F	X	2607	CopyOnWrite	0
Total								
Track(s)		823715						
MB(s)		25741.0						

Legend:

(G): The Target device is associated with this group.

Example 6: Obtaining more device group information

The `symdg list` command displays a list of any device groups that have already been defined in your host's SYMAPI database file or GNS-managed global repository. The `symdg show DgName` command displays the membership and configuration details of any existing device group.¹ This example illustrates options for obtaining device group information not described earlier.

This display is similar to the `symdg show RDFGrp1` display described previously, but shows a configuration in which five local R1 devices are the source devices and five remotely associated BCVs are the targets. That is, the R1 devices copy data to their R2 devices, which copy that data to the BCVs located on the same remote Symmetrix array.

symdg show rdf1-grp

Group Name: rdf1-grp

```

Group Type                : RDF1
Device Group in GNS       : Yes
Valid                     : Yes
Symmetrix ID              : 000187900035
Group Creation Time       : Tue Jan  6 12:08:17 2006
Vendor ID                 : EMC Corp
Application ID            : SYMCLI

```

```

Number of STD Devices in Group : 5
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 0
Number of Locally-associated VDEV's : 0
Number of Remotely-associated BCV's (STD RDF) : 5
Number of Remotely-associated BCV's (BCV RDF) : 0
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 0

```

Standard (STD) Devices (5):

```

{
-----
LdevName          PdevName          Sym   Att.  Sts   Cap
Dev              Dev              (MB)
-----
DEV001            /dev/vx/rdmp/c4t0d102s2 03C2   RW    4315
DEV002            /dev/vx/rdmp/c4t0d103s2 03C3   RW    4315
DEV003            /dev/vx/rdmp/c4t0d104s2 03C4   RW    4315
DEV004            /dev/vx/rdmp/c4t0d105s2 03C5   RW    4315
DEV005            /dev/vx/rdmp/c4t0d106s2 03C6   RW    4315
}

```

BCV Devices Remotely-associated (STD RDF) (5):

```

{
-----
LdevName          PdevName          Sym   Att.  Sts   Cap
Dev              Dev              (MB)
-----
RBCV001           N/A              02E0   RW    4315
RBCV002           N/A              02E1   RW    4315
RBCV003           N/A              02E2   RW    4315
RBCV004           N/A              02E3   RW    4315
RBCV005           N/A              02E4   RW    4315
}

```

1. The `symcg list` and `symcg show` commands display the same information for a composite group.

This second part of the `syndg show` output displays the device group's RDF information, including the identity of the remote Symmetrix array (sid 41), RDF configuration settings (Enabled or Disabled), and the combined state of all standard RDF devices in the group and their RDF links.

```

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

  Remote Symmetrix ID                     : 000187900041

  R2 Device Is Larger Than The R1 Device : False

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

  RDF Device Domino                       : Disabled

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

  Device RDF Status                       : Ready             (RW)

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

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

  Device RDF State                        : Ready             (RW)
  Remote Device RDF State                 : Write Disabled     (WD)

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

  Number of R1 Invalid Tracks             : 0
  Number of R2 Invalid Tracks             : 12
}

```

It is possible to get device configuration and status information about devices by querying the device group to which they have been added. The `symmir query` command displays the status of BCV pairs in a device group, `Rdf1Grp`, that contains one SRDF standard device and one local BCV device. The BCV pair is currently in the Split state.

symmir -g Rdf1Grp query

Device Group (DG) Name: Rdf1Grp
 DG's Type : RDF1
 DG's Symmetrix ID : 000185500810

Standard Device			BCV Device			State
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. Tracks	STD <=> BCV
DEV001	010F	0	BCV001	00CD *	0	Split
Total		-----			-----	
MB(s)		0.0			0.0	

Legend:

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

The `symrdf query` command displays the state of the SRDF devices and their RDF links. Under normal circumstances, the SRDF pair is synchronized. The RDF link is operating with synchronous replication (indicated by an S in the M column).

symrdf -g Rdf1Grp query

Device Group (DG) Name: Rdf1Grp
 DG's Type : RDF1
 DG's Symmetrix ID : 000000003264

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
		ST			LI	ST					
Standard		A			N	A					
Logical		T	R1 Inv	R2 Inv	K	T	R1 Inv	R2 Inv	RDF Pair		
Device	Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	MDA	STATE
-----					-----					-----	
DEV001	009C	RW	0	0	RW	0054	WD	0	0	S..	Synchronized
DEV002	009D	RW	0	0	RW	0055	WD	0	0	S..	Synchronized
Total		-----			-----			-----			
Track(s)		0			0			0			0
MB(s)		0.0			0.0			0.0			0.0

Legend for MODES:

M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino) : X = Enabled, . = Disabled
 A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off

The `symclone query` command with the `-multi` option displays a source device and its four clone targets. The state of the clone operation shows that all four copy operations are currently in progress (CopyInProgress) and that each operation is 6% complete.

symclone -g multiclone query -multi

Device Group (DG) Name: multiclone
 DG's Type : REGULAR
 DG's Symmetrix ID : 000187900041

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
CLONESRC	0215	129275	CLONETGT1	0100	XX	CopyInProgress	6
		129277	CLONETGT2	0101	XX	CopyInProgress	6
		129276	CLONETGT3	0102	XX	CopyInProgress	6
		129276	CLONETGT4	0103	XX	CopyInProgress	6
Total		-----					
Track(s)		517104					
MB(s)		16159.5					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The `symdbg show` command with the `-lock` option displays existing device external locks on devices within the device group named `sar`. The display shows which devices on the local and remote Symmetrix arrays are locked during Symmetrix Automated Replication (that is, in the middle of a SRDF/AR cycle).

symdbg show sar -lock

S Y M M E T R I X D E V I C E G R O U P L O C K S

Group Name: sar

Group Type : REGULAR
 Valid : Yes
 Symmetrix ID : 000185500814
 Group Creation Time : Tue Oct 14 09:49:41 2003
 Vendor ID : EMC Corp
 Application ID : SYMCLI
 Lock Number : 9

Number of Locked Devices in Group:

STD Devices	:	16	(SID: 000185500814)
Remote Mirrors of STD's	:	0	(SID: N/A)
BCV Devices	:	16	(SID: 000185500814)
Remote Mirrors of BCV's	:	16	(SID: 000185500505)
RBCV Devices	:	0	(SID: N/A)
Remote Mirrors of RBCV's	:	0	(SID: N/A)
BRBCV Devices	:	16	(SID: 000185500505)
Remote Mirrors of BRBCV's	:	0	(SID: N/A)

STD Devices (16):

Sym	Lock	Director	Seconds	Lock
Dev	Holder-ID	Initiator	Number	Held

				Flags

```

03F7 0x1001100E 0 05B 12 Long
03F8 0x1001100E 0 05B 12 Long
03F9 0x1001100E 0 05B 12 Long
03FA 0x1001100E 0 05B 12 Long
03FB 0x1001100E 0 05B 12 Long
03FC 0x1001100E 0 05B 12 Long
03FD 0x1001100E 0 05B 12 Long
03FE 0x1001100E 0 05B 12 Long
03FF 0x1001100E 0 05B 12 Long
0400 0x1001100E 0 05B 12 Long
0401 0x1001100E 0 05B 12 Long
0402 0x1001100E 0 05B 12 Long
0403 0x1001100E 0 05B 12 Long
0404 0x1001100E 0 05B 12 Long
0405 0x1001100E 0 05B 12 Long
0406 0x1001100E 0 05B 12 Long
}

```

BCV Devices (16):

```

{
-----
Sym   Lock      Director  Seconds  Lock
Dev   Holder-ID Initiator  Number   Held    Flags
-----
0407  0x1001100E 0        05B      13      Long
0408  0x1001100E 0        05B      13      Long
0409  0x1001100E 0        05B      13      Long
040A  0x1001100E 0        05B      13      Long
040B  0x1001100E 0        05B      13      Long
040C  0x1001100E 0        05B      13      Long
040D  0x1001100E 0        05B      13      Long
040E  0x1001100E 0        05B      13      Long
040F  0x1001100E 0        05B      13      Long
0410  0x1001100E 0        05B      13      Long
0411  0x1001100E 0        05B      13      Long
0412  0x1001100E 0        05B      13      Long
0413  0x1001100E 0        05B      13      Long
0414  0x1001100E 0        05B      13      Long
0415  0x1001100E 0        05B      13      Long
0416  0x1001100E 0        05B      13      Long
}

```

Remote Mirrors of BCV's (16):

```

{
-----
Sym   Lock      Director  Seconds  Lock
Dev   Holder-ID Initiator  Number   Held    Flags
-----
0390  0x100113A3 0        14B      13      Long
0391  0x100113A3 0        14B      13      Long
0392  0x100113A3 0        14B      13      Long
0393  0x100113A3 0        14B      13      Long
0394  0x100113A3 0        14B      13      Long
0395  0x100113A3 0        14B      13      Long
0396  0x100113A3 0        14B      13      Long
0397  0x100113A3 0        14B      13      Long
0398  0x100113A3 0        14B      13      Long
0399  0x100113A3 0        14B      13      Long
039A  0x100113A3 0        14B      13      Long
039B  0x100113A3 0        14B      13      Long
039C  0x100113A3 0        14B      13      Long
039D  0x100113A3 0        14B      13      Long
039E  0x100113A3 0        14B      13      Long
039F  0x100113A3 0        14B      13      Long
}

```

BRBCV Devices (16):

```

{
-----
Sym   Lock      Director  Seconds  Lock
Dev   Holder-ID  Initiator Number   Held    Flags
-----
03A0  0x100113A3  0         14B      13      Long
03A1  0x100113A3  0         14B      13      Long
03A2  0x100113A3  0         14B      13      Long
03A3  0x100113A3  0         14B      13      Long
03A4  0x100113A3  0         14B      13      Long
03A5  0x100113A3  0         14B      13      Long
03A6  0x100113A3  0         14B      13      Long
03A7  0x100113A3  0         14B      13      Long
03A8  0x100113A3  0         14B      13      Long
03A9  0x100113A3  0         14B      13      Long
03AA  0x100113A3  0         14B      13      Long
03AB  0x100113A3  0         14B      13      Long
03AC  0x100113A3  0         14B      13      Long
03AD  0x100113A3  0         14B      13      Long
03AE  0x100113A3  0         14B      13      Long
03AF  0x100113A3  0         14B      13      Long
}

```

Example 7: Obtaining disk and audit information

This example describes some of the more relevant displays from `syndisk` and `symaudit` commands. The `syndisk list` command displays the configuration and status of the physical disks and their hypervolumes within a Symmetrix array. A disk is identified by its DA symbol (Symb), its interface (Int) and its target ID (TID). To display verbose information about a specific disk (for example, 01A:C3) and its hypervolumes, use `syndisk list -da 01A -interface C -tid 3 -v -hypers`.

syndisk list

```
Symmetrix ID      : 000185500814
Disks Selected    : 96
```

Ident	Symb	Int	TID	Vendor	Type	Hypers	Capacity (MB)	
							Total	Free
DA-1A	01A	C	0	SEAGATE	CHET_73	17	70007	446
DA-1A	01A	C	1	SEAGATE	CHET_73	17	70007	446
DA-1A	01A	C	2	SEAGATE	CHET_73	16	70007	962
DA-1A	01A	C	3	SEAGATE	CHET_73	16	70007	962
DA-1A	01A	C	4	SEAGATE	CHET_73	15	70007	5465
DA-1A	01A	C	5	SEAGATE	CHET_73	19	70007	5977
DA-1A	01A	D	0	SEAGATE	CHET_73	17	70007	446
DA-1A	01A	D	1	SEAGATE	CHET_73	17	70007	446
DA-1A	01A	D	2	SEAGATE	CHET_73	16	70007	962
DA-1A	01A	D	3	SEAGATE	CHET_73	16	70007	962
DA-1A	01A	D	4	SEAGATE	CHET_73	17	70007	1666
DA-1A	01A	D	5	SEAGATE	CHET_73	16	70007	2697
DA-2A	02A	C	0	SEAGATE	CHET_73	17	70007	446
DA-2A	02A	C	1	SEAGATE	CHET_73	17	70007	446
DA-2A	02A	C	2	SEAGATE	CHET_73	16	70007	962
DA-2A	02A	C	3	SEAGATE	CHET_73	16	70007	962
DA-2A	02A	C	4	SEAGATE	CHET_73	16	70007	5466
DA-2A	02A	C	5	SEAGATE	CHET_73	19	70007	5977
DA-2A	02A	D	0	SEAGATE	CHET_73	17	70007	446
DA-2A	02A	D	1	SEAGATE	CHET_73	16	70007	962
DA-2A	02A	D	2	SEAGATE	CHET_73	16	70007	962
DA-2A	02A	D	3	SEAGATE	CHET_73	18	70007	3612
DA-2A	02A	D	4	SEAGATE	CHET_73	18	70007	1667
DA-2A	02A	D	5	SEAGATE	CHET_73	18	70007	1667
DA-15A	15A	C	0	SEAGATE	CHET_73	17	70007	446
DA-15A	15A	C	1	SEAGATE	CHET_73	17	70007	446
DA-15A	15A	C	2	SEAGATE	CHET_73	16	70007	962
DA-15A	15A	C	3	SEAGATE	CHET_73	16	70007	962
DA-15A	15A	C	4	SEAGATE	CHET_73	16	70007	5466
DA-15A	15A	C	5	SEAGATE	CHET_73	19	70007	1664
DA-15A	15A	D	0	SEAGATE	CHET_73	17	70007	446
DA-15A	15A	D	1	SEAGATE	CHET_73	16	70007	962
DA-15A	15A	D	2	SEAGATE	CHET_73	16	70007	962
DA-15A	15A	D	3	SEAGATE	CHET_73	16	70007	962

The `symaudit` commands retrieve information from the Symmetrix Audit Log file from data written to the file during control operations. This information allows you to determine what application on what host initiated actions that directed Symmetrix behavior. The `symaudit show` command below displays the time period and quantity of data in the Audit Log file that relate to local Symmetrix 814.

symaudit show

```

                A U D I T   L O G   D A T A

Symmetrix ID           : 000185500814

Starting date          : 12/14/2003 14:53:10
Ending date            : 12/17/2003 14:40:47

Starting record number :      1
Ending record number   :     127
Total record count     :     127

```

The following `symaudit list` command displays the extracted audit log records for local Symmetrix 814 during the time period that data was collected. The ellipsis (.....) indicates where some output was omitted.

symaudit list

```

                A U D I T   L O G   D A T A

Symmetrix ID           : 000185500814

Starting date          : 12/14/2003 14:53:10
Ending date            : 12/17/2003 14:40:47

Record
Number  Date      Time      Application      Host      Function Class  Action Code
-----
      1 12/14/03 14:53:10  SYMCLI          api179     RDF      Split
      2 12/14/03 14:53:10  SYMCLI          api179     RDF      Split
      3 12/14/03 14:53:11  SYMCLI          api179     RDF      Split
      4 12/14/03 14:53:16  SYMCLI          api179     RDF      IncEsta
      5 12/14/03 14:53:16  SYMCLI          api179     RDF      IncEsta
      6 12/14/03 14:53:17  SYMCLI          api179     RDF      IncEsta
.....
     14 12/14/03 16:32:39  SYMCLI          api179     BCV      FullEsta
     15 12/14/03 16:32:39  SYMCLI          api179     BCV      FullEsta
     16 12/14/03 16:33:01  SYMCLI          api179     BCV      FullEsta
     17 12/14/03 16:40:01  SYMCLI          api179     BCV      Split
     18 12/14/03 16:40:01  SYMCLI          api179     BCV      Split
     19 12/14/03 16:40:01  SYMCLI          api179     BCV      Split
.....
    109 12/17/03 14:39:05  SYMCLI          api179     BCV      Split
    110 12/17/03 14:39:07  SYMCLI          api179     RDF      IncEsta
    111 12/17/03 14:39:07  SYMCLI          api179     RDF      IncEsta
    112 12/17/03 14:39:08  SYMCLI          api179     RDF      IncEsta
    113 12/17/03 14:39:10  SYMCLI          api179     BCV      IncEsta
    114 12/17/03 14:39:10  SYMCLI          api179     BCV      IncEsta
    115 12/17/03 14:39:10  SYMCLI          api179     BCV      IncEsta
    116 12/17/03 14:39:10  SYMCLI          api179     BCV      IncEsta
    117 12/17/03 14:39:24  SYMCLI          api179     BCV      IncEsta
    118 12/17/03 14:39:24  SYMCLI          api179     BCV      IncEsta
    119 12/17/03 14:39:24  SYMCLI          api179     BCV      IncEsta
    120 12/17/03 14:39:24  SYMCLI          api179     BCV      IncEsta
    121 12/17/03 14:39:24  SYMCLI          api179     BCV      IncEsta
    122 12/17/03 14:39:42  SYMCLI          api179     BCV      IncEsta
    123 12/17/03 14:40:44  SYMCLI          api179     BCV      Split
    124 12/17/03 14:40:44  SYMCLI          api179     BCV      Split

```

```

125 12/17/03 14:40:44 SYMCLI      api179      BCV      Split
126 12/17/03 14:40:44 SYMCLI      api179      BCV      Split
127 12/17/03 14:40:46 SYMCLI      api179      BCV      Split

```

Using the `-text` option with the `symaudit list` command displays the text associated with the audit log record.

symaudit list -text

```

          A U D I T   L O G   D A T A

Symmetrix ID           : 000185500814

Starting date          : 12/14/2003 14:53:10
Ending date            : 12/17/2003 14:40:47

Record   Time   Function   Action
Number   -----   Class     Code
-----
      Text
-----
  1 14:53:10 RDF      Split
    STARTING an RDF 'SPLIT'. type: Regular [RDF1 BCV], flags:

  2 14:53:10 RDF      Split
    [ 0407-0416 ]

  3 14:53:11 RDF      Split
    The RDF 'SPLIT' operation SUCCEEDED.

  4 14:53:16 RDF      IncEsta
    STARTING an RDF 'INCREMENTAL ESTABLISH'. type: Regular [RDF1 BCV], flags:

  5 14:53:16 RDF      IncEsta
    [ 0407-0416 ]

  6 14:53:17 RDF      IncEsta
    The RDF 'INCREMENTAL ESTABLISH' operation SUCCEEDED.
-----
 14 16:32:39 BCV      FullEsta
    STARTING a local 'BCV Full Establish' operation for 1 [STD-BCV] Pair:

 15 16:32:39 BCV      FullEsta
    [ 0000-0190 ]

 16 16:33:01 BCV      FullEsta
    'BCV Full Establish'. sts: SYMAPI_C_SUCCESS

 17 16:40:01 BCV      Split
    STARTING a local 'BCV Split' operation for 1 [STD-BCV] Pair:
 18 16:40:01 BCV      Split
    [ 0000-0190 ]

 19 16:40:01 BCV      Split
    'BCV Split'. sts: SYMAPI_C_SUCCESS
-----
109 14:39:05 BCV      Split
    'BCV Split'. sts: SYMAPI_C_SUCCESS

110 14:39:07 RDF      IncEsta
    STARTING an RDF 'INCREMENTAL ESTABLISH'. type: Regular [RDF1 BCV], flags:

111 14:39:07 RDF      IncEsta
    [ 0407-0416 ]

112 14:39:08 RDF      IncEsta

```

```

The RDF 'INCREMENTAL ESTABLISH' operation SUCCEEDED.

113 14:39:10 BCV      IncEsta
      STARTING a local 'BCV Inc Establish' operation for 16 [STD-BCV] Pairs:

114 14:39:10 BCV      IncEsta
      [ 03F7-0407 03F8-0408 03F9-0409 03FA-040A 03FB-040B 03FC-040C

115 14:39:10 BCV      IncEsta
      03FD-040D 03FE-040E 03FF-040F 0400-0410 0401-0411 0402-0412

116 14:39:10 BCV      IncEsta
      0403-0413 0406-0416 0404-0415 0405-0414 ]

117 14:39:24 BCV      IncEsta
      'BCV Inc Establish'. sts: SYMAPI_C_SUCCESS

118 14:39:24 BCV      IncEsta
      STARTING a remote 'BCV Inc Establish' operation for 16 [STD-BCV] Pairs:

119 14:39:24 BCV      IncEsta
      [ 0390-03A0 0391-03A1 0392-03A2 0393-03A3 0394-03A4 0395-03A5

120 14:39:24 BCV      IncEsta
      0396-03A6 0397-03A7 0398-03A8 0399-03A9 039A-03AA 039B-03AB

121 14:39:24 BCV      IncEsta
      039C-03AC 039F-03AF 039E-03AE 039D-03AD ]

122 14:39:42 BCV      IncEsta
      'BCV Inc Establish'. sts: SYMAPI_C_SUCCESS

123 14:40:44 BCV      Split
      STARTING a remote 'BCV Split' operation for 16 [STD-BCV] Pairs:

124 14:40:44 BCV      Split
      [ 0390-03A0 0391-03A1 0392-03A2 0393-03A3 0394-03A4 0395-03A5

125 14:40:44 BCV      Split
      0396-03A6 0397-03A7 0398-03A8 0399-03A9 039A-03AA 039B-03AB

126 14:40:44 BCV      Split
      039C-03AC 039F-03AF 039E-03AE 039D-03AD ]

127 14:40:46 BCV      Split
      'BCV Split'. sts: SYMAPI_C_SUCCESS

```

The following `symaudit list` command displays detailed information about record number 124 during a TimeFinder split operation (that is, the BCV is being split from its paired standard device). The Text line displays which BCV pairs are being split (for example, device 390 from device 3A0).

```
symaudit -v list -record_num 124
```

```

      A U D I T   L O G   D A T A

Symmetrix ID           : 000185500814

Starting date          : 12/17/2003 14:40:44
Ending date            : 12/17/2003 14:40:47

Record Number          :      124
Records in Seq         :      4
Offset in Seq          :      2
Time                   : 12/17/03 14:40:44
Vendor ID              : EMC Corp
Application ID         : SYMCLI
Application Version    : 5.4.0.0
API Library            : SEK
API Version            : T5.4.0.0 (Edit Level: 236)
Host Name              : api179
OS Name                : SunOS
OS Revision            : 5.8Generic
Client Host            :
Process ID             : 00006670
Task ID                : 00000000
Function Class         : BCV
Action Code            : Split
Text                   : [ 0390-03A0 0391-03A1 0392-03A2 0393-03A3 0394-03A4 0395-03A5

```

Example 8: Obtaining display output in XML format

You can specify an XML output mode for SYMCLI utilities outputs using an environment variable setting or the `-out xml` option on the command line. Valid modes are `Standard`, `XML`, or `XML_ELEMENT`. The example uses the `symcfg list` command to illustrate output.

By default, the `symcfg list` command returns a standard output as shown below.

symcfg list

```

                                S Y M M E T R I X

SymmID      Attachment  Model      Mcode      Cache      Num Phys      Num Symm
              Version    Size (MB)   Devices     Devices

000000006164 Local        2000P-M2   5671        32768        125         647
000187900039 Local        800-M2     5671        24576        104         425
000187940256 Local        800-M2     5671        32768        94          507

```

The `export` command sets the `SYMCLI_OUTPUT_MODE` variable. By setting this global variable, all SYMCLI commands will return output in the format specified (when possible). You can set it to `Standard` (default), `XML` (attribute-based), or `XML_ELEMENT` (element-based). For example, to set it to `XML_ELEMENT`, enter:

export SYMCLI_OUTPUT_MODE=xml_ELEMENT

The `symcfg list` command now produces the same output in element-based XML format.

symcfg list

```

<?xml version="1.0" standalone="yes" ?>
<SymCLI_ML>
  <Symmetrix>
    <Symm_Info>
      <symid>000000006164</symid>
      <attachment>Local</attachment>
      <model>2000P-M2</model>
      <microcode_version>5671</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <physical_devices>125</physical_devices>
      <devices>647</devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000187900039</symid>
      <attachment>Local</attachment>
      <model>800-M2</model>
      <microcode_version>5671</microcode_version>
      <cache_megabytes>24576</cache_megabytes>
      <physical_devices>104</physical_devices>
      <devices>425</devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000187940256</symid>
      <attachment>Local</attachment>
      <model>800-M2</model>
      <microcode_version>5671</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <physical_devices>94</physical_devices>
      <devices>507</devices>
    </Symm_Info>
  </Symmetrix>
</SymCLI_ML>

```

The `unset` command terminates the previous setting for `SYMCLI_OUTPUT_MODE`. In effect, this operation returns the mode setting to standard mode.

unset SYMCLI_OUTPUT_MODE

By default, the `symcfg list` command should now produce its output in standard mode. However, you can specify an output mode on the command line using the `-out` `<Standard|XML|XML_ELEMENT>` option. This allows you to set the output for a specific operation, to again display output in either XML format or standard format.

symcfg list -out xml_element

```
<?xml version="1.0" standalone="yes" ?>
<SymCLI_ML>
  <Symmetrix>
    <Symm_Info>
      <symid>000000006164</symid>
      <attachment>Local</attachment>
      <model>2000P-M2</model>
      <microcode_version>5671</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <physical_devices>125</physical_devices>
      <devices>647</devices>
    </Symm_Info>
  </Symmetrix>
</Symmetrix>
  <Symm_Info>
    <symid>000187900039</symid>
    <attachment>Local</attachment>
    <model>800-M2</model>
    <microcode_version>5671</microcode_version>
    <cache_megabytes>24576</cache_megabytes>
    <physical_devices>104</physical_devices>
    <devices>425</devices>
  </Symm_Info>
</Symmetrix>
</Symmetrix>
  <Symm_Info>
    <symid>000187940256</symid>
    <attachment>Local</attachment>
    <model>800-M2</model>
    <microcode_version>5671</microcode_version>
    <cache_megabytes>32768</cache_megabytes>
    <physical_devices>94</physical_devices>
    <devices>507</devices>
  </Symm_Info>
</Symmetrix>
</SymCLI_ML>
```

Example 9: Obtaining status information about GNS

Group Name Services (GNS) propagates SYMAPI device group and composite group definitions across multiple hosts and Symmetrix arrays. GNS stores device group and composite group definitions in a shared repository that is accessible from multiple hosts. Thus, you can perform SYMCLI and SYMAPI control operations from any host in the Symmetrix environment. If you make a change to a group definition on one host, that change is visible to other hosts.

The `stordaeon list` command displays whether the GNS and watchdog daemons are running on your host.

stordaeon list

Available Daemons ('[*]': Currently Running):

storapid	EMC Solutions Enabler Base Daemon
[*] storgnsd	EMC Solutions Enabler GNS Daemon
[*] storwatchd	EMC Solutions Enabler Watchdog Daemon
stororad	EMC Solutions Enabler SRM Oracle Daemon
storudbd	EMC Solutions Enabler SRM UDB Daemon
storifmxd	EMC Solutions Enabler SRM Informix Daemon
storsybs11d	EMC Solutions Enabler SRM Sybase 11 Daemon
storsybs12d	EMC Solutions Enabler SRM Sybase 12 Daemon
storsybs12.5d	EMC Solutions Enabler SRM Sybase 12.5 Daemon
storsybs12.5_64d	EMC Solutions Enabler SRM Sybase 12.5 Daemon, 64-bit

The `symdg list` command with the `-inactive` option displays device groups that exist currently in your host's SYMAPI database. From the GNS-enabled viewpoint, these are inactive device groups.

symdg list -inactive

I N A C T I V E D E V I C E G R O U P S								
Name	Type	Valid	Symmetrix ID	Devs	Number of			
					GKs	BCVs	VDEVs	
79api179regular	REGULAR	N/A	000187700079	23	0	3	0	
35api179regular	REGULAR	N/A	000187900035	22	0	44	0	

The `symdg activateall` command imports all device groups from your host's SYMAPI database into GNS. From the GNS-enabled viewpoint, these are now active device groups

symdg activateall -noprompt

Activating Device Group: '79api179regular' ... Succeeded.
Activating Device Group: '35api179regular' ... Succeeded.

If GNS is enabled on your host, the `symdg list` command displays device groups that are active from the GNS-enabled viewpoint.

symdg list

D E V I C E G R O U P S								
Name	Type	Valid	Symmetrix ID	Devs	Number of			
					GKs	BCVs	VDEVs	
79api179regular	REGULAR	Yes	000187700079	23	0	3	0	
35api179regular	REGULAR	Yes	000187900035	22	0	44	0	

This chapter provides real-world examples of using Access Control:

- ◆ Example 1: Access Control group, pool, and ACE creation 14-2
- ◆ Example 2: Access Control and TimeFinder 14-6
- ◆ Example 3: Access Control and SRDF 14-7
- ◆ Example 4: Access Control and configuration manager 14-12
- ◆ Example 5: Access Control and older EMC software products 14-22
- ◆ Example 6: Access Control and EMC ControlCenter 14-24

Example 1: Access Control group, pool, and ACE creation

In the following example, the host in AdminGrp is admin, and the host that is being administered is HRSun1. The Symmetrix array on which Access Control is being regulated has the Symmetrix ID 000185400077. Access Control has been enabled at the Symmetrix service processor for Symmetrix ID 000185400077.

Examine the Access Control status from admin host. The display shows the default Access Control status after being enabled.

```
admin# symacl list -acl -sid 77
```

Symmetrix ID: 000185400077

Group Name	Pool Name	Access Type
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

Examine the Access Control status from the HRSun1 host.

```
HRSun1# symacl list -acl
```

Symmetrix ID: 000185400077

Group Name	Pool Name	Access Type
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

The UnknwGrp access group has ALL access rights, so all operations (except administrative tasks) are allowed. Examine the devices. The ellipsis (...) indicates where output was omitted.

```
HRSun1# symdev list
```

Symmetrix ID: 000185400077

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute	Sts
016 /dev/rdisk/c1t0d18s2	16A:0 01B:C5 BCV	Asst'd	RW 480
017 /dev/rdisk/c1t0d19s2	16A:0 02A:D0 BCV	N/Asst'd	RW 480
018 /dev/rdisk/c1t0d20s2	16A:0 02A:C5 BCV	N/Asst'd	RW 480
019 /dev/rdisk/c1t0d21s2	16A:0 01B:D0 BCV	N/Asst'd	RW 480
03F /dev/rdisk/c1t0d51s2	16A:0 01B:C5 Unprotected	N/Grp'd	RW 480
040 /dev/rdisk/c1t0d52s2	16A:0 02A:C5 Unprotected	N/Grp'd	RW 480
041 /dev/rdisk/c1t0d53s2	16A:0 01A:C5 Unprotected	N/Grp'd	RW 480
042 /dev/rdisk/c1t0d54s2	16A:0 02B:D4 Unprotected	N/Grp'd	RW 480

Pick two standard devices (03F and 040) and two BCVs (017 and 018) and create a device group.

```
HRSun1# symdg create acltest
HRSun1# symld -g acltest addall dev -range 3F:40
HRSun1# symbcv -g acltest associateall dev -range 17:18
HRSun1# symld -g acltest list
```

Device Group (DG) Name: acltest

DG's Type : REGULAR
 DG's Symmetrix ID : 000185400077

Standard Device Name			Directors			Device		
Logical	Physical	Sym	SA	:P DA	:IT Config	Att	Sts	Cap (MB)
DEV001	c1t0d51s2	03F +	16A:0	01B:C5	Unprotected	RW		480
DEV002	c1t0d52s2	040 +	16A:0	02A:C5	Unprotected	RW		480

Legend for STD devices:

(+): Paired with a BCV device that is associated with this dg.

(-): Paired with a BCV device that is non-associated with this dg.

BCV Devices associated with this dg:

BCV Device				Standard Device			Status
Logical	Sym	RDF Att.	Inv. Tracks	Logical	Sym	Inv. Tracks	BCV <=> STD
BCV001	017	+	0	DEV001	03F	0	Split
BCV002	018	+	0	DEV002	040	0	Split
Total							
Track(s)				0			
MB(s)				0.0			

Legend for BCV devices:

(+): BCV is paired with a member STD device.

(-): BCV is paired with a non-member STD device.

Now assign the HRSun1 host to the HRGRP access group, allowing it only BASE access rights. A file `acl.cmd` containing the appropriate Access Control commands has already been created on the admin host.

```
admin# cat acl.cmd
```

```
create accgroup HRGRP;
Add host accid 2F5B07A6-5C408FCB-9C3A757A name HRSun1 to accgroup HRGRP;
Create accpool HRPOOL;
Add dev 3F:40 to accpool HRPOOL; Add dev 17:18 to accpool HRPOOL;
Grant access=BASE to accgroup HRGRP for ALL DEVS;
```

Execute the file on the admin host and verify the results. Use of the `preview` and `prepare` arguments is recommended, although they are not shown here (to limit output). The PIN must be entered unless the environment variable `SYMCLI_ACCESS_PIN` is set.

```
admin# symacl -sid 77 -file acl.cmd commit
```

Enter Access PIN:

Command file: (acl.cmd)

PREVIEW.....Started.

PREVIEW.....Done.

PREPARE.....Started.

Creating group HRGRP.....Done.

Adding Host access id api131 to group HRGRP.....Done.

Creating pool HRPOOL.....Done.

```

Adding 2 devices, devices 03F through 040 to pool HRPOOL.....Done.

Adding 2 devices, devices 017 through 018 to pool HRPOOL.....Done.

Granting access BASE to group HRGRP for ALL devices.....Done.

PREPARE.....Done.

Starting COMMIT.....Done.

Confirm that the HRSun1 host has been added to the newly created HRGRP access group.
admin# symacl show accgroup HRGRP -sid 77

```

Access Group: HRGRP

Symmetrix ID: 000185400077

```

Access ID Name
-----
HRSun1

```

Confirm that the HRPOOL device pool contains the devices that were placed in it. The devices are not visible from the admin host.

```
admin# symacl -sid 77 show accpool HRPOOL
```

Access Pool: HRPOOL

```

Symmetrix ID           : 000185400077
Number of Member Devices : 4

```

Member Devices (6):

```

{
    Device Name           Device
    -----
Sym Physical           Config      Attribute  Sts      Cap
-----
017 Not Visible        BCV          N/Asst'd   NR       480
018 Not Visible        BCV          N/Asst'd   NR       480
03F Not Visible        Unprotected  N/Grp'd    RW       480
040 Not Visible        Unprotected  N/Grp'd    RW       480
}

```

A listing of all ACLs shows that the HRGRP access group has BASE access rights but no BCV access rights.

```
admin# symacl list -acl -sid 77
```

Symmetrix ID: 000185400077

```

Group Name      Pool Name      Access Type
-----
AdminGrp        ALL_DEVS        ADMIN
AdminGrp        ALL_DEVS        ALL
UnknwGrp        ALL_DEVS        BASE
UnknwGrp        !INPOOLS        ALL
HRGRP           ALL_DEVS        BASE

```

A view from the HRSun1 host shows only the ACLs associated with the access group to which the HRSun1 host belongs.

```
HRSun1# symacl list -acl -sid 77
```

Symmetrix ID: 000185400077

Group Name	Pool Name	Access Type
-----	-----	-----
HRGRP	ALL_DEVS	BASE

The HRSun1 host can also see the HRPOOL device pool.

```
HRSun1# symacl show accpool HRPOOL -sid 77
```

Access Pool: HRPOOL

```
Symmetrix ID           : 000185400077
Number of Member Devices : 4
```

Member Devices (6):

```
{
    Device Name
    -----
    Sym Physical          Config      Attribute  Sts      Cap
    -----
    017 /dev/rdisk/c1t0d19s2  BCV        Asst'd     NR       480
    018 /dev/rdisk/c1t0d20s2  BCV        Asst'd     NR       480
    03F /dev/rdisk/c1t0d51s2  Unprotected Grp'd      RW       480
    040 /dev/rdisk/c1t0d52s2  Unprotected Grp'd      RW       480
}
```

With BASE access rights, the HRSun1 host can get information from the Symmetrix array.

```
HRSun1# symmir query -g acltest
```

```
Device Group (DG) Name: acltest
DG's Type           : REGULAR
DG's Symmetrix ID    : 000185400077
```

Standard Device			BCV Device		State
Logical	Inv. Sym Tracks	Logical	Inv. Sym Tracks	STD <=>	BCV
DEV001	03F 0	BCV001	017 *	0	Split
DEV002	040 0	BCV002	018 *	0	Split
Total	-----			-----	
Track(s)	0			0	
MB(s)	0.0			0.0	

Legend:

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

However, a TimeFinder control operation is denied, because the HRSun1 host does not have BCV access rights.

```
HRSun1# symmir -g acltest establish -noprompt
```

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

Symmetrix access control denied the request

Example 2: Access Control and TimeFinder

Use the admin host to interactively assign the BCV access right to the HRGRP access group.

```
admin# setenv SYMCLI_ACCESS_PIN xxxx
admin# symacl -sid 77 commit <<EOF
Grant access=BCV to accgroup HRGRP for accpool HRPOOL;
EOF

Command file: (stdin)
PREVIEW.....Started.
PREVIEW.....Done.

PREPARE.....Started.

Granting access BCV to group HRGRP for pool HRPOOL.....Done.

PREPARE.....Done.

Starting COMMIT.....Done.
```

```
admin# symacl list -sid 77 -acl
```

```
Symmetrix ID: 000185400077
```

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL
HRGRP	ALL_DEVS	BASE
HRGRP	HRPOOL	BCV

TimeFinder control operations now execute correctly on the HRSun1 host.

```
HRSun1# symmir -g acltest establish -noprompt
```

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

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

Example 3: Access Control and SRDF

A second Symmetrix array with ID 000185400124, along with Symmetrix ID 000185400077, is used in this example. Access Control has been enabled at the service processor for Symmetrix ID 000185400124. The admin host from the previous example has been given ADMIN access rights to the target Symmetrix array.

Examine the Access Control status on the target Symmetrix array. The display shows the default Access Control status after being enabled.

```
admin# symacl -sid 124 list -acl
```

```
Symmetrix ID: 000185400124
```

Group Name	Pool Name	Access Type
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

On the HRSun1 host, build an RDF1 device group consisting of two devices (001 and 002). Again, the ellipsis (...) indicates where output was omitted.

```
HRSun1# symdev list
```

```
Symmetrix ID: 000185400077
```

Device Name	Directors	Device
Sym Physical	SA :P DA :IT Config	Attribute Sts Cap (MB)
000 Not Visible	??? :? 02A:D0 RDF1	N/Grp'd RW 480
001 /dev/rdisk/clt0d1s2	16A:0 01B:C5 RDF1	N/Grp'd RW 480
002 /dev/rdisk/clt0d2s2	16A:0 01A:D0 RDF1	N/Grp'd RW 480
003 /dev/rdisk/clt0d3s2	16A:0 02B:C5 RDF1	N/Grp'd RW 480
004 /dev/rdisk/clt0d4s2	16A:0 02B:C0 RDF1	N/Grp'd RW 480

```
HRSun1# symdbg create acltest2 -type rdf1
```

```
HRSun1# symld -g acltest2 addall dev -range 001:002
```

```
HRSun1# symrdf -g acltest2 query
```

```
Device Group (DG) Name: acltest2
DG's Type : RDF1
DG's Symmetrix ID : 000185400077
```

Source (R1) View				Target (R2) View				M O D E S			
Standard	ST	A	LI	ST	A	M					
Logical	T R1 Inv	R2 Inv	K	T R1 Inv	R2 Inv	d				RDF Pair	
Device	Dev	E Tracks	Tracks	S Dev	E Tracks	Tracks	e	Dom	ACp	STATE	
DEV001	001 RW	0	0 NR 03D WD	0	0	0	0	SYN	DIS	OFF	Suspended
DEV002	002 RW	0	0 NR 03E WD	0	0	0	0	SYN	DIS	OFF	Suspended
Total											
Track(s)		0	0		0	0					
MB(s)		0.0	0.0		0.0	0.0					

The `symdev list` and `symrdf query` commands have worked because the HRSun1 host has BASE access rights on the local Symmetrix array (77). On the remote Symmetrix array, the HRSun1 host is part of the UnknwGrp access group, since its access ID is not registered as part of a defined access group. An RDF control command is denied to the HRSun1 host because the HRGRP access group on Symmetrix 77 does not have RDF access rights.

```
HRSun1# symrdf establish -g acltest2 -noprompt
```

An RDF 'Incremental Establish' operation execution is in progress for device group 'acltest2'. Please wait...

Symmetrix access control denied the request

Interactively add the RDF access right to the HRGRP access group.

```
admin# symacl -sid 77 commit <<EOF
Add dev 01:02 to accpool HRPOOL;
Grant access=RDF to accgroup HRGRP for accpool HRPOOL;
EOF
```

```
Command file: (stdin)
PREVIEW.....Started.
PREVIEW.....Done.

PREPARE.....Started.

Adding 2 devices, devices 001 through 002 to pool HRPOOL...Done.

Granting access RDF to group HRGRP for pool HRPOOL.....Done.

PREPARE.....Done.

Starting COMMIT.....Done.
```

Examining the ACLs for Symmetrix 77 shows that the RDF access right has been granted to the HRGRP access group.

```
admin# symacl -sid 77 show accgroup HRGRP -acl
```

Symmetrix ID: 000185400077

Group Name	Pool Name	Access Type
-----	-----	-----
HRGRP	ALL_DEVS	BASE
HRGRP	HRPOOL	BCV
HRGRP	HRPOOL	RDF

The two RDF devices also show up in the HRPOOL device pool.

```
admin# symacl show accpool HRPOOL -sid 77
```

Access Pool: HRPOOL

```
Symmetrix ID           : 000185400077
Number of Member Devices : 6
```

Member Devices (6):

Device Name	Device
-----	-----
Sym Physical	Config Attribute Sts Cap (MB)
-----	-----
001 Not Visible	RDF1 N/Grp'd RW 480
002 Not Visible	RDF1 N/Grp'd RW 480
017 Not Visible	BCV N/Asst'd NR 480

```

018 Not Visible          BCV          N/Asst'd      NR          480
03F Not Visible          Unprotected N/Grp'd      RW          480
040 Not Visible          Unprotected N/Grp'd      RW          480
}

```

The RDF control operation from the HRSun1 host now executes correctly.

```
HRSun1# symrdf -g acltest2 establish -noprompt
```

An RDF 'Incremental Establish' operation execution is in progress for device group 'acltest2'. Please wait...

```

Suspend RDF link(s).....Done.
Resume RDF link(s).....Not Done.
Merge device track tables between source and target.....Started.
Devices: 001-002 ..... Merged.
Merge device track tables between source and target.....Done.
Resume RDF link(s).....Done.

```

The RDF 'Incremental Establish' operation successfully initiated for device group 'acltest2'.

An RDF operation requires access to both the source and target devices. If access to the target devices 03D and 03E on Symmetrix 124 is removed from the UnknwGrp access group, the HRSun1 host can no longer perform RDF operations on the device group acltest2.

```
admin# symacl list -acl -sid 124
```

Symmetrix ID: 000185400124

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

Remove the BASE access right from the UnknwGrp access group on the target Symmetrix ID 000185400124.

```

admin# symacl -sid 124 commit <<EOF
remove access=BASE from accgroup UnknwGrp for ALL DEVS;
EOF

```

```

Command file: (stdin)
PREVIEW.....Started.
PREVIEW.....Done.

PREPARE.....Started.

  Removing access BASE from group UnknwGrp for ALL devices...Done.

PREPARE.....Done.

Starting COMMIT.....Done.

```

```
admin# symacl -sid 124 show accgroup UnknwGrp -acl
```

Symmetrix ID: 000185400124

Group Name	Pool Name	Access Type
-----	-----	-----
UnknwGrp	!INPOOLS	ALL

A symrdf query from the HRSun1 host cannot provide information on the remote Symmetrix array, because the HRSun1 host is not permitted to get any information from the remote Symmetrix array.

```
HRSun1# symrdf -g acltest2 query
```

```
Device Group (DG) Name: acltest2
DG's Type           : RDF1
DG's Symmetrix ID    : 000185400077
```

Source (R1) View					Target (R2) View					M O D E S			
-----					-----					-----			
Standard	ST				LI	ST				M			
Logical	A				N	A				o			
Device	T	R1 Inv	R2 Inv	K	T	R1 Inv	R2 Inv	d				RDF Pair	
Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	e	Dom	ACp	STATE	
-----					-----					-----			
DEV001	001	RW	0	0	NR	03D	NA	NA	NA	SYN	DIS	OFF	Partitioned
DEV002	002	RW	0	0	NR	03E	NA	NA	NA	SYN	DIS	OFF	Partitioned
Total		-----		-----	-----		-----						
Track(s)		0		0	NA		NA						
MB(s)		0.0		0.0	NA		NA						

As expected, an RDF control operation fails.

```
HRSun1# symrdf -g acltest2 split -noprompt
```

```
An RDF 'Split' operation execution is in progress for device group 'acltest2'.
Please wait...
```

```
Symmetrix access control denied the request
```

Successful queries from the HRSun1 host can be performed by reinstating the BASE access right to the UnknwGrp access group.

```
admin# symacl -sid 124 commit<<EOF
grant access=BASE to accgroup UnknwGrp for ALL DEVS;
EOF
```

```
Command file: (stdin)
PREVIEW.....Started.
PREVIEW.....Done.

PREPARE.....Started.

Granting access BASE to group UnknwGrp for ALL devices.....Done.

PREPARE.....Done.

Starting COMMIT.....Done.
```

```
admin# symacl -sid 124 show accgroup UnknwGrp -acl
```

```
Symmetrix ID: 000185400124
```

Group Name	Pool Name	Access Type
UnknwGrp	!INPOOLS	ALL
UnknwGrp	ALL_DEVS	BASE

The symrdf query command now works from the HRSun1 host.

```
HRSun1# symrdf -g acltest2 query
```

```
Device Group (DG) Name: acltest2
DG's Type           : RDF1
DG's Symmetrix ID   : 000185400077
```

Source (R1) View				Target (R2) View				M O D E S			
-----				-----				-----			
Standard	ST			LI	ST			M			
	A			N	A			o			
Logical	T R1 Inv	R2 Inv		K	T R1 Inv	R2 Inv		d			RDF Pair
Device	Dev	E Tracks	Tracks	S Dev	E Tracks	Tracks		e	Dom	ACp	STATE
-----				-----				-----			
DEV001	001 RW	0	0	RW 03D WD	0	0	0	SYN DIS OFF	Synchronized		
DEV002	002 RW	0	0	RW 03E WD	0	0	0	SYN DIS OFF	Synchronized		
Total											
Track(s)		0	0			0	0				
MB(s)		0.0	0.0			0.0	0.0				

But a symrdf split operation does not work.

```
HRSun1# symrdf -g acltest2 -noprompt split
```

```
An RDF 'Split' operation execution is in progress for device group 'acltest2'.
Please wait...
```

```
Symmetrix access control denied the request
```

Example 4: Access Control and configuration manager

This example outlines the SYMCLI commands and Access Control operations that are required to reconfigure devices on a Symmetrix array. Refer to the *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* for further information on configuring devices.

Two new devices on host `cfg_host` are created from the free disk space on the Symmetrix array. The new devices are converted into a meta device, and then mapped to a front-end port, correcting the port flags for the new host type that is attached to this port.

This procedure requires the host performing the commands to have the following access rights: BASE, BASECTRL, CFGDEV, CFGSYM, CREATEDV, and SDR.

Table 14-1 Example: Required access rights

Required access rights	Description of permissions
BASE	Required to view the free space on the Symmetrix array
CREATEDV	Required to create the two new Symmetrix devices with the correct protection
CFGDEV	Required to convert the two Symmetrix devices to a meta device
CFGSYM	Required to set the port flags for the new host
SDR	Require to map the meta device to a front-end port
BASECTRL	Required to make the meta device Not Ready

In the example, the host in the AdminGrp access group is `admin`, and the host that is being administered is `cfg_host`. The Symmetrix array on which Access Control is being regulated has the Symmetrix ID 000184600040. Access Control has been enabled at the Symmetrix service processor for Symmetrix ID 000184600040.

Set up the access rights. The command file `acl.txt` on the admin host contains the following Access Control commands:

```
create accgroup cfg_grp;
add host accid 2F5B07A6-ABCEDF12-9A3D757A name cfg_host to accgroup cfg_grp;
grant access=BASE,CREATEDV,CFGDEV to accgroup cfg_grp for ALL DEVS;
grant access=BASECTRL,SDR,CFGSYM to accgroup cfg_grp for ALL DEVS;
```

Execute the command file on the admin host to set up access rights for the `cfg_host` host.

```
admin# symacl -sid 40 commit -file acl.txt
```

Enter Access PIN:

Command file: (acl.txt)

```
PREVIEW.....Started.
PREVIEW.....Done.
```

```
PREPARE.....Started.
```

```
Creating group cfg_grp.....Done.
```

```
Adding Host access id cfg_host to group cfg_grp.....Done.
```

```
Granting access CFGDEV to group cfg_grp for ALL devices.....Done.
```

```
Granting access CREATEDV to group cfg_grp for ALL devices.....Done.
```

```
Granting access BASE to group cfg_grp for ALL devices.....Done.
```

```

Granting access CFGSYM to group cfg_grp for ALL devices.....Done.
Granting access SDR to group cfg_grp for ALL devices.....Done.
Granting access BASECTRL to group cfg_grp for ALL devices.....Done.
PREPARE.....Done.
Starting COMMIT.....Done.

```

A listing of all ACLs from the admin host confirms the access rights for the cfg_host host.

```
admin# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

View the ACLs from the cfg_host host.

```
cfg_host# export SYMCLI_SID=000184600040
```

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

Create a meta device:

Prepare to create two new devices by checking for free disk space. The default output is in cylinders.

```
cfg_host# symconfigure -freespace list
```

```
Symmetrix ID: 000184600040
```

Emulation	Formatted (Cyl)	Unformatted (Cyl)
-----	-----	-----
FBA	2732468	0

List devices before creating the two new devices. Notice that the last device in the listing is device 012.

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute Sts	
000 /dev/rdisk/c1t0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd WD	8
001 /dev/rdisk/c1t0d1s2	03B:0 02B:C1 Unprotected	N/Grp'd RW	3
002 /dev/rdisk/c1t0d2s2	03B:0 01A:C2 Unprotected	N/Grp'd RW	3
003 Not Visible	??? :? 01A:C0 Unprotected	N/Grp'd RW	1875
004 Not Visible	??? :? 02B:C3 Unprotected	N/Grp'd RW	1875
005 Not Visible	??? :? 02A:C0 Unprotected	N/Grp'd RW	1875
006 Not Visible	??? :? 01B:C3 Unprotected	N/Grp'd RW	1875
007 Not Visible	??? :? 01B:C0 2-Way Mir	N/Grp'd (M) RW	7500
008 Not Visible	??? :? 02B:C0 2-Way Mir	N/Grp'd (m) RW	-
009 Not Visible	??? :? 01A:D0 2-Way Mir	N/Grp'd (m) RW	-
00A Not Visible	??? :? 02A:D0 2-Way Mir	N/Grp'd (m) RW	-
00B Not Visible	??? :? 01B:D0 BCV	N/Asst'd RW	1875
00C Not Visible	??? :? 02A:D2 BCV	N/Asst'd RW	1875
00D /dev/rdisk/c1t0d13s2	03B:0 02B:D0 2-Way BCV Mir	N/Asst'd RW	1875
00E /dev/rdisk/c1t0d14s2	03B:0 01A:C1 2-Way BCV Mir	N/Asst'd RW	1875
00F Not Visible	??? :? 02A:C1 DRV	N/Grp'd RW	1875
010 Not Visible	??? :? 01B:C2 DRV	N/Grp'd RW	1875
011 Not Visible	??? :? 01A:D1 2-Way Mir	N/A (FS) RW	2878
012 Not Visible	??? :? 02A:D1 2-Way Mir	N/A (FS) RW	2878

Create two new devices. This operation will fail if the access rights do not include CREATDV. In this example, the two new devices are 1024 cylinders in size (which equals 480 MB).

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symconfigure commit <<EOF
```

```
> create dev count=2, size=1024, emulation=FBA, config=2-Way-Mir;
> EOF
```

```
A Configuration Change operation is in progress. Please wait...
Establishing a configuration change session.....Established.
Processing symmetrix 000185400123
Submitting configuration changes.....Submitted.
Validating configuration changes.....Validated.
Initiating PREPARE of configuration changes.....Queued.
PREPARE requesting required resources.....Obtained.
Step 006 of 010 steps.....Executing.

Step 009 of 010 steps.....Executing.
Local: PREPARE.....Done.
Initiating COMMIT of configuration changes.....Queued.
COMMIT requesting required resources.....Obtained.
```

```
Step 004 of 097 steps.....Executing.
```

```
Step 097 of 097 steps.....Executing.
```

```
Local: COMMIT.....Done.
```

```
Terminating the configuration change session.....Done.
```

The configuration change session has successfully completed.

When you add new devices, the SYMAPI host database file on the host issuing the configuration change is updated automatically on successful completion of the `symconfig commit` command. (Automatic update of the local SYMAPI database occurs for all configuration changes except mapping changes. To refresh the SYMAPI host database files on all other hosts attached to that Symmetrix array, you can perform the `symcfg sync` command on those hosts.)

List the devices. Notice the two new devices, 013 and 014. Each of these is 480 MB.

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute	Sts
000 /dev/rdisk/clt0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd	WD 8
012 Not Visible	???? 02A:D1 2-Way Mir	N/A (FS)	RW 2878
013 Not Visible	???? 02B:C3 2-Way Mir	N/Grp'd	RW 480
014 Not Visible	???? 01A:C0 2-Way Mir	N/Grp'd	RW 480

Convert the two new devices into a meta device. This operation will fail if the access rights do not include CFGDEV.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symconfigure commit <<EOF
> form meta from dev 13, config=concatenated;
> add dev 14 to meta 13;
> EOF
```

A Configuration Change operation is in progress. Please wait...

```
Establishing a configuration change session.....Established.
Processing symmetrix 000184600040
```

```
Local: COMMIT.....Done.
```

```
Terminating the configuration change session.....Done.
```

The configuration change session has successfully completed.

List the devices again to check that the two devices have been converted into a meta device.

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute	Sts
000 /dev/rdisk/c1t0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd	WD 8
013 Not Visible	???:? 02B:C3 2-Way Mir	N/Grp'd (M) RW	960
014 Not Visible	???:? 01A:C0 2-Way Mir	N/Grp'd (m) RW	-

Check the port flags. These must be reset for a new host. In this example, a new host is connected to port 03A:0, and the newly created meta device is added to the port. The old host connected to this port was running HP-UX and required special port settings (Disk Array enabled and Volume Set Addressing enabled). The new host requires these port flags to be disabled.

```
cfg_host# symcfg -SA 03A -P 0 -v list
```

```
Symmetrix ID: 000184600040 (Local)
```

```

Product Model           : 8130
Symmetrix ID            : 000184600040

Microcode Version (Number) : 5567 (15BFAA01)
Microcode Date           : 06.11.2001

Microcode Patch Date     : 06.11.2001
Microcode Patch Level    : 30

```

Fibre Specific Flags

```

{
  Disk_Array           : Enabled
  Volume_Set_Addressng : Enabled
  Hard_Addressng       : Disabled
  Non_Participating    : Disabled
  Global_3rdParty_Logout : Disabled
  Init_Point_to_Point  : Enabled
  Unique_WWN           : Enabled
  Generic_VSA          : Disabled
  VCM_Enabled          : Enabled
  Class_2_Service      : Disabled
  OpenVMS              : Disabled
}

```

Change the port flags. This operation will fail if the access rights do not include CFGSYM.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symconfigure commit <<EOF
>set port 03A:0 Disk_Array=DISABLE, Volume_Set_Addresssing=DISABLE;
>EOF
```

A Configuration Change operation is in progress. Please wait...

```
Establishing a configuration change session.....Established.
Processing symmetrix 000184600040
```

```
.....
Local: COMMIT.....Done.
Terminating the configuration change session.....Done.
```

The configuration change session has successfully completed.

Check the port settings.

```
cfg_host# symcfg -SA 03A -P 0 -v list
```

```
Symmetrix ID: 000184600040 (Local)
```

```
.....
Fibre Specific Flags
{
  Disk_Array           : Disabled
  Volume_Set_Addresssing : Disabled
  Hard_Addresssing     : Disabled
  Non_Participating    : Disabled
  Global_3rdParty_Logout : Disabled
  Init_Point_to_Point   : Enabled
  Unique_WWN           : Enabled
  Generic_VSA          : Disabled
  VCM_Enabled          : Enabled
  Class_2_Service      : Disabled
  OpenVMS              : Disabled
}
```

The port settings are now set correctly for the new host.

Map the meta device to a port. The mapping operation requires specifying a LUN address. Use the `symcfg list -addresses` command to display the fibre channel addresses to determine the next available LUN address.

```
cfg_host# symcfg -sa 3A -address list
```

```
Symmetrix ID: 000184600040 (Local)
```

Director			Device Name		Address		
Ident	Symbolic	Port	Sym	Physical	VBUS	TID	LUN
FA-3A	03A	0	000	Not Visible	0	0	000
			001	Not Visible	0	0	001
			002	Not Visible	0	0	002
			003	Not Visible	0	0	003
			004	Not Visible	0	0	004
			005	Not Visible	0	0	005
			006	Not Visible	0	0	006
			007	Not Visible	0	0	007
			00B	Not Visible	0	0	00B
			00C	Not Visible	0	0	00C
			00D	Not Visible	0	0	00D
			00E	Not Visible	0	0	00E

The LUN address needed for the new meta device is 00F. Map the meta device. This operation will fail if the access rights do not include SDR.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symconfigure commit <<EOF
> map dev 13 to dir 03A:0 lun=0F;
> EOF
```

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session.....Established.

The `symdev list` command displays that the mapping operation was successful, indicated by the presence now of 03A:0 in the SA:P column of the meta head (013) and meta tail (014). The meta members will continue to be marked “Not Visible” until you execute the host update utilities that assign the host names.

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name		Directors		Device		
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)
000	/dev/rdisk/clt0d0s2	03B:0 01B:C1	2-Way Mir	N/Grp'd	WD	8
013	Not Visible	03A:0 02B:C3	2-Way Mir	Grp'd (M)	RW	960
014	Not Visible	03A:0 01A:C0	2-Way Mir	N/Grp'd (m)	RW	-

The meta device is now mapped and the port is set correctly. Instead of updating the host, this example now proceeds directly to unmapping the meta device.

Before unmapping any device, you need to ensure that there is no host I/O to the device. In this example, a device group is created, the meta device is added to the group, and then the device is made Not Ready. You can use the `symdev` command to make a device Not Ready for any I/O. Recall that the meta head (device 013) represents the entire meta device.

```
cfg_host# symdg create meta_dev_grp
```

```
cfg_host# symld -g meta_dev_grp add dev 13
```

```
cfg_host# symdg show meta_dev_grp
```

```
Group Name: meta_dev_grp
```

```

Group Type           : REGULAR
Valid                : Yes
Symmetrix ID         : 000184600040
Group Creation Time  : Mon Jun 11 18:14:31 2001
Vendor ID            : EMC Corp
Application ID       : SYMCLI

```

```

Number of STD Devices in Group : 1
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 0
Number of Remotely-associated BCV's (STD RDF): 0
Number of Remotely-associated BCV's (BCV RDF): 0

```

```
Standard (STD) Devices (1):
```

```
{
```

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
DEV001	N/A	013	(M)	RW	960

```
}
```

Make the meta device Not Ready. This operation will fail if the access rights do not include BASECTRL.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symld -g meta_dev_grp not_ready DEV001
```

```
Execute a 'Not Ready' Device operation for device 'DEV001' in device group
'meta_dev_grp' (y/[n]) ? y
```

```
'Not Ready' Device operation successfully completed for the device.
```

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
-----	-----	-----	-----
Sym Physical	SA :P DA :IT Config	Attribute	Sts
-----	-----	-----	-----
000 /dev/rdisk/c1t0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd	WD 8
-----	-----	-----	-----
013 Not Visible	03A:0 02B:C3 2-Way Mir	Grp'd (M) NR	960
014 Not Visible	03A:0 01A:C0 2-Way Mir	N/Grp'd (m) NR	-

Unmap the meta device. This operation will fail if the access rights do not include SDR.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symconfigure commit <<EOF
```

```
> unmap dev 13 from dir 03A:0;
```

```
> EOF
```

```
A Configuration Change operation is in progress. Please wait...
```

```
Establishing a configuration change session.....Established.
```

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute	Sts
000 /dev/rdisk/clt0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd	WD 8
013 Not Visible	???:? 02B:C3 2-Way Mir	Grp'd (M) NR	960
014 Not Visible	???:? 01A:C0 2-Way Mir	N/Grp'd (m) NR	-

Make the meta device Ready. This operation will fail if the access rights do not include BASECTRL.

```
cfg_host# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
cfg_grp	ALL_DEVS	BASE
cfg_grp	ALL_DEVS	CREATEDV
cfg_grp	ALL_DEVS	CFGDEV
cfg_grp	ALL_DEVS	SDR
cfg_grp	ALL_DEVS	BASECTRL
cfg_grp	ALL_DEVS	CFGSYM

```
cfg_host# symld -g meta_dev_grp ready DEV001
```

Execute a 'Ready' Device operation for device 'DEV001' in device group
'meta_dev_grp' (y/[n]) ? **y**

'Ready' Device operation successfully completed for the device.

```
cfg_host# symdev list
```

```
Symmetrix ID: 000184600040
```

Device Name	Directors	Device	Cap (MB)
Sym Physical	SA :P DA :IT Config	Attribute	Sts
000 /dev/rdisk/clt0d0s2	03B:0 01B:C1 2-Way Mir	N/Grp'd	WD 8
013 Not Visible	???:? 02B:C3 2-Way Mir	Grp'd (M) RW	960
014 Not Visible	???:? 01A:C0 2-Way Mir	N/Grp'd (m) RW	-

Example 5: Access Control and older EMC software products

EMC software products developed before Access Control was implemented, and software products that do not use the SYMAPI, are not aware of Access Control. Software calls to the Symmetrix array from these products do not include the access ID. Access Control views such software calls as originating from an unknown host (that is, as not belonging to a member of any access group), and operates according to the access rights defined for the UnknwGrp access group. These software products also do not abide by Access Control device pool definitions, and therefore Access Control cannot restrict the devices that they use.

There are two products that do not use the SYMAPI and which will be blocked by access control rights. The EMC ControlCenter V4.3 (or older) SDR GUI component is one. This product needs the SDR access right to all devices (for the UnknwGrp access group) to properly commit the configuration change to the Symmetrix array. The ESN Manager, Volume Logix component (V2.0 or older) is the other product that does not use the SYMAPI, and is used in this example to illustrate the operation of a software product developed before Access Control was implemented.

In this example, the Volume Logix `fpath` command writes to and updates the Volume Logix database only if the UnknwGrp access group has the ALL access right, or the VLOGIX and BASE access rights.

By default, Access Control permits any host connected to the Symmetrix array to execute Volume Logix control commands. This is because the UnknwGrp access group has ALL rights to all non-pooled devices, and immediately after Access Control has been deployed there are no user defined device pools.

```
admin# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

Restrict the privileges of the UnknwGrp access group to demonstrate the rejection of Volume Logix control commands. This is accomplished by removing the ALL access right to all non-pooled devices.

```
admin# symacl -sid 40 commit<<EOF
```

```
remove access=ALL from accgroup UnknwGrp for NON-POOLED DEVS;
EOF
```

```
Command file: (stdin)
PREVIEW.....Started.
PREVIEW.....Done.
```

```
PREPARE.....Started.
```

```
Removing access ALL from group UnknwGrp for NON-POOLED devs..Done.
```

```
PREPARE.....Done.
```

```
Starting COMMIT.....Done.
```

```
admin# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE

The fpath commands will fail because access rights for Volume Logix have been removed for the UnknwGrp access group. Note that the error messages do not refer to an access control violation; Volume Logix was developed before Access Control was implemented.

```
admin# C:\Program Files\EMC\Volume Logix4>fpath adddev -d \\.\PHYSICALDRIVE1
-w 10000000C9245401 -f 16A -r 7
```

```
Copyright EMC Corp. 1998, 1999, 2000 and 2001 @(#) fpath Revision: 2.4.4 $
```

```
fpath adddev "FAILED write to VCM database failed"
```

```
admin# C:\Program Files\EMC\Volume Logix5>fpath swaphba -d \\.\PHYSICALDRIVE1
-w 10000000C9245401 -n 10000000C926AF80
```

```
Copyright EMC Corp. 1998, 1999, 2000 and 2001 @(#) fpath Revision: 2.5.1 $
```

```
fpath swaphba "FAILED write to VCM database failed"
```

```
admin# C:\Program Files\EMC\Volume Logix5>fpath clrwwn -d \\.\PHYSICALDRIVE1
-w 10000000C9245401
```

```
Copyright EMC Corp. 1998, 1999, 2000 and 2001 @(#) fpath Revision: 2.5.1 $
```

```
fpath clrwwn "FAILED write to VCM database failed"
```

The ability to perform Volume Logix fpath commands can be re-instated by assigning the ALL privilege to the UnknwGrp access group for all devices. By doing so, the UnknwGrp access group will continue to be able to control devices that might be placed into user defined pools at a later time.

```
admin# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
-----	-----	-----
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL
UnknwGrp	ALL_DEVS	VLOGIX
New_Grp	ALL_DEVS	BASE
New_Grp	New_POOL	ALL

Example 6: Access Control and EMC ControlCenter

EMC ControlCenter allows a user at a graphical console to manage a Symmetrix array through use of locally installed host agents (ControlCenter agents) that send SYMAPI commands to the Symmetrix array. ControlCenter is Access Control aware, and Symmetrix actions issued by ControlCenter agents are subject to Access Control. A specific access right (ECC) is provided for ControlCenter applications.

Examine the Access Control status from the admin host. The display shows the default status after Access Control has been enabled on the Symmetrix array.

```
admin# symacl list -acl
```

```
Symmetrix ID: 000184600040
```

Group Name	Pool Name	Access Type
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL

The default settings created during Access Control initialization allow ControlCenter agents to run on the Symmetrix array. All hosts have the ALL access right to all devices (they are either in the AdminGrp, or they are part of the UnknwGrp, which has the ALL access right to all devices, since device pools are not defined immediately after Access Control is enabled), so they are considered to have the ControlCenter access right. The ControlCenter access PIN is assumed to be part of the UnknwGrp.

However, as you create access groups and device pools and customize the default Access Control settings for your environment, you must follow the procedure outlined in this example, or follow the directions listed in the ControlCenter online help, in order to use ControlCenter applications.

Set up the access group for ControlCenter applications. The only supported access rights for ControlCenter applications are the ALL or BASE access rights. Create a command file (here called `SetUpECC.txt`) on the admin host containing the following Access Control commands. (Comment lines in the command file start with the characters `##`. The ControlCenter access PIN is excluded from this documentation; it is only available to purchasers of ControlCenter.)

```
## Create an access group for EMC ControlCenter applications.
## This is called ECC_APP.
create accgroup ECC_APP;
## Add an access PIN and name for ControlCenter applications
add user accid <Access PIN> name ECCPin to accgroup ECC_APP;
## Grant ALL (or BASE) access to all devices for ControlCenter applications
grant access=ALL to accgroup ECC_APP for ALL DEVS;
```

Execute the command file to set up access rights for ControlCenter applications.

```
admin# symacl -sid 40 commit -file SetUpECC.txt
```

```
Enter Access PIN:
Command file: (SetUpECC.txt)
```

```
PREVIEW.....Started.
PREVIEW.....Done.
```

A listing of all ACLs confirms the access rights for ControlCenter applications.

```
admin# symacl list -acl
```

```
Symmetrix ID: 000018450040
```

Group Name	Pool Name	Access Type
AdminGrp	ALL_DEVS	ADMIN
AdminGrp	ALL_DEVS	ALL
UnknwGrp	ALL_DEVS	BASE
UnknwGrp	!INPOOLS	ALL
ECC_APP	ALL_DEVS	ALL

Each host that will run ControlCenter agents must have the ControlCenter access right (ECC). To do this, add the ControlCenter access right to an existing access group that contains the ControlCenter agent host. Alternatively, set up a separate access group that contains all the ControlCenter agent hosts, and give that group the BASE and ECC access rights, as is done here. Create a command file (here called `SetUpECC_Hosts.txt`) on the admin host containing the following Access Control commands.

```
## Create an access group called ECC_HOST for hosts running ControlCenter
## agents
create accgroup ECC_HOST;
## Add an access ID to the group for each host running a ControlCenter
## agent, where:
##     <UNIQUE ID> is the unique ID of the host returned by running
##     "symacl -unique" on that host
##     <HOST NAME> is the name of that host
add host accid <UNIQUE ID> name <HOST NAME> to accgroup ECC_HOST;
grant access=BASE,ECC to accgroup ECC_HOST for ALL DEVS;
```

Execute the command file on the admin host to set up access rights for all hosts that will run ControlCenter agents. For example, insert the appropriate access ID and host name for a single ControlCenter agent host `ECC_host` in the command file, and execute the command file on the admin host.

```
admin# symacl -sid 40 commit -file SetUpECC_Hosts.txt
```

```
Enter Access PIN:
```

```
Command file: (SetUpECC_Hosts.txt)
```

```
PREVIEW.....Started.
PREVIEW.....Done.
```

A listing of all ACLs from the ControlCenter agent host confirms the access rights for that host. The display does not contain any information on the access rights given to the ControlCenter application; this can be displayed only from the admin host.

```
ECC_host# symacl list -acl
```

```
Symmetrix ID: 000018450040
```

Group Name	Pool Name	Access Type
ECC_HOST	ALL_DEVS	BASE
ECC_HOST	ALL_DEVS	ECC

The ControlCenter application, which is run from a ControlCenter agent, has ALL access rights to all devices in the Symmetrix array. (It is recommended that the right to run ControlCenter agents should be limited to a small number of secure hosts.) Other applications on the ControlCenter agent host that use the SYMAPI, or SYMCLI commands issued from the

ControlCenter agent host, have only BASE access rights, because that is the only access right assigned to the ECC_HOST access group other than the ControlCenter (ECC) access right.

The ControlCenter application has its own security authentication and authorization mechanisms that provide control over which ControlCenter functions can be performed from the ControlCenter Console. You can also create rules in ControlCenter to allow users to perform actions on groups of devices. This allows a ControlCenter Administrator to limit the use of a ControlCenter Console. Refer to the *EMC ControlCenter User Guide* for a complete description of the ControlCenter authentication and authorization systems.

This chapter provides realworld examples of how to use GNS.

- ◆ Example 1: Importing device groups into GNS15-2
- ◆ Example 2: Using GNS to propagate a composite group definition15-6

Example 1: Importing device groups into GNS

The following example was performed using Solutions Enabler version 5.4. Two device groups defined in the SYMAPI database of one host are imported into GNS, which then propagates these device group definitions to another locally attached host running GNS.

The following command uses the vi text editor to open the SYMAPI options file and set the option SYMAPI_USE_GNS to ENABLE. This action enables GNS for all applications on the host.

```
vi /var/symapi/config/options

SYMAPI_USE_GNS=ENABLE
```

The following command uses the vi text editor to open the daemon options file and set the option AUTORESTART to ENABLE. Doing this makes use of the watchdog daemon, which automatically restarts the GNS daemon if it crashes. (Beginning with Solutions Enabler version 6.0 and Symmetrix arrays running Enginuity version 5x71, this option is enabled by default.)

```
vi /var/symapi/config/daemon_options

storgnsd:autorestart=enabled
```

The `stordaeomon start` command starts the GNS daemon manually. To start this daemon automatically, refer to the section "Starting the GNS Daemon."

```
stordaeomon start storgnsd
```

Waiting for daemon to start. This may take several seconds.

The `stordaeomon list` command verifies that the GNS and watchdog daemons are running on the host.

```
stordaeomon list
```

Available Daemons ('[*]': Currently Running):

```

    storapid                EMC Solutions Enabler Base Daemon
[*] storgnsd                EMC Solutions Enabler GNS Daemon
[*] storwatchd             EMC Solutions Enabler Watchdog Daemon

    stororad                EMC Solutions Enabler SRM Oracle Daemon
    storudbd                EMC Solutions Enabler SRM UDB Daemon
    storifmxd               EMC Solutions Enabler SRM Informix Daemon
    storsybs11d              EMC Solutions Enabler SRM Sybase 11 Daemon
    storsybs12d              EMC Solutions Enabler SRM Sybase 12 Daemon
    storsybs12.5d            EMC Solutions Enabler SRM Sybase 12.5 Daemon
    storsybs12.5_64d         EMC Solutions Enabler SRM Sybase 12.5 Daemon, 64-bit
```

The `symdg list` command with the `-inactive` option displays device groups that exist currently in the host's SYMAPI database. From the GNS-enabled viewpoint, these are inactive device groups.

```
symdg list -inactive
```

I N A C T I V E D E V I C E G R O U P S								
Name	Type	Valid	Symmetrix ID	Devs	Number of			
					GKs	BCVs	VDEVs	
79api179regular	REGULAR	N/A	000187700079	23	0	3	0	
35api179regular	REGULAR	N/A	000187900035	22	0	44	0	

The `symdg activateall` command imports all device groups from the SYMAPI database into GNS. From the GNS-enabled viewpoint, these are now active device groups.

```
symdg activateall -noprompt
```

```
Activating Device Group: '79api179regular' ... Succeeded.
Activating Device Group: '35api179regular' ... Succeeded.
```

The `symdg list` command displays device groups from the GNS-enabled viewpoint.

```
symdg list
```

D E V I C E G R O U P S							
Name	Type	Valid	Symmetrix ID	Devs	Number of		
					GKs	BCVs	VDEVs
79api179regular	REGULAR	Yes	000187700079	23	0	3	0
35api179regular	REGULAR	Yes	000187900035	22	0	44	0

The `symdg show` command displays details of the device group named 79api179regular.

```
symdg show 79api179regular
```

```
Group Name: 79api179regular
```

```
Group Type                : REGULAR
Device Group in GNS       : Yes
Valid                     : Yes
Symmetrix ID              : 000187700079
Group Creation Time       : Thu Nov 20 16:14:16 2003
Vendor ID                 : EMC Corp
Application ID            : SYMCLI
```

```
Number of STD Devices in Group : 23
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 3
Number of Locally-associated VDEV's : 0
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 0
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 0
```

```
Standard (STD) Devices (23):
```

```
{
-----
LdevName                PdevName                Sym   Dev  Att.  Sts   Cap
                        Dev                               (MB)
-----
DEV001                  /dev/vx/rdmp/c5t0d6s2  0042  RW    2878
DEV002                  /dev/vx/rdmp/c5t0d7s2  0043  RW    2878
DEV003                  /dev/vx/rdmp/c5t0d8s2  0044  RW    2878
DEV004                  /dev/vx/rdmp/c5t0d9s2  0045  RW    2878
DEV005                  /dev/vx/rdmp/c5t0d10s2 0046  RW    2878
DEV006                  /dev/vx/rdmp/c5t0d11s2 0047  RW    2878
DEV007                  /dev/vx/rdmp/c5t0d12s2 0048  RW    2878
DEV008                  /dev/vx/rdmp/c5t0d13s2 0049  RW    2878
DEV009                  /dev/vx/rdmp/c5t0d14s2 004A  RW    2878
DEV010                  /dev/vx/rdmp/c5t0d15s2 004B  RW    2878
DEV011                  /dev/vx/rdmp/c5t0d16s2 004C  RW    2878
DEV012                  /dev/vx/rdmp/c5t0d17s2 004D  RW    2878
DEV013                  /dev/vx/rdmp/c5t0d18s2 004E  RW    2878
DEV014                  /dev/vx/rdmp/c5t0d19s2 004F  RW    2878
DEV015                  /dev/vx/rdmp/c5t0d20s2 0050  RW    2878
DEV016                  /dev/vx/rdmp/c5t0d21s2 0051  RW    2878
```

```

DEV017          /dev/vx/rdump/c5t0d22s2  0052      RW      2878
DEV018          /dev/vx/rdump/c5t0d23s2  0053      RW      2878
DEV019          /dev/vx/rdump/c5t0d24s2  0054      RW      2878
DEV020          /dev/vx/rdump/c5t0d25s2  0348 (M)   RW      14391
DEV021          /dev/vx/rdump/c5t0d26s2  034D (M)   RW      14391
DEV022          /dev/vx/rdump/c5t0d27s2  0370 (M)   RW      14391
DEV023          /dev/vx/rdump/c5t0d28s2  0375 (M)   RW      14391
}

```

BCV Devices Locally-associated (3):

```

{
-----
LdevName          PdevName          Sym      Att.  Sts      Cap
Dev              (MB)
-----
BCV001            N/A              0454      NR      2878
BCV002            N/A              0455      RW      2878
BCV003            N/A              0456      RW      2878
}

```

Device Group BCV RDF Information

```

{
RDF Type                      : R1
RDF (RA) Group Number         : 3                      (02)

Remote Symmetrix ID           : 000187700067

R2 Device Is Larger Than The R1 Device : False

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

RDF Device Domino             : Disabled

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

Device RDF Status              : Ready                  (RW)

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

Device Suspend State           : N/A
Device Consistency State       : Disabled
RDF R2 Not Ready If Invalid    : Enabled

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

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

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

```

The example now enables GNS on another host connected to the same Symmetrix array by setting the SYMAPI_USE_GNS option in this host's SYMAPI options file as was shown previously. After starting the GNS daemon on this host, you can use the `symdg list` command to verify that GNS has propagated the device group definitions to this host.

`symdg list`

D E V I C E			G R O U P S				
Name	Type	Valid	Symmetrix ID	Devs	Number of		
					GKs	BCVs	VDEVs
79api179regular	REGULAR	Yes	000187700079	23	0	3	0
35api179regular	REGULAR	Yes	000187900035	22	0	44	0

Example 2: Using GNS to propagate a composite group definition

The following example was performed using Solutions Enabler V5.4. The hardware setup has two hosts (Host A and Host B) connected to three Symmetrix arrays (SIDs 43, 60, and 35). GNS is enabled and running on both hosts. PowerPath is installed on both hosts.

From Host A, the example creates a composite group definition within GNS, which automatically adds the definition to the PowerPath database on each host. The example then updates the composite group definition from Host B by adding additional devices. GNS automatically propagates the updated composite group definition to the PowerPath database on Host A.

From Host A, the `symcg create` command creates an RDF1 composite group with the name 3-sym-gns. Because GNS is enabled on the host, the group is stored in GNS, which automatically adds it to the PowerPath database on Host A and Host B.

```
symcg create 3-sym-gns -type rdf1 -ppath
```

The `symcg addall` commands add to the composite group a range of devices from two Symmetrix arrays (SIDs 43 and 60).

```
symcg -cg 3-sym-gns addall dev -range 110:128 -sid 43  
symcg -cg 3-sym-gns addall dev -range 784:788 -sid 60
```

The `symcg enable` command enables the composite group for RDF consistency protection, which transforms the group into a consistency group.

```
symcg -cg 3-sym-gns enable -noprompt
```

A consistency 'Enable' operation execution is in progress for composite group '3-sym-gns'. Please wait...

The consistency 'Enable' operation successfully executed for composite group '3-sym-gns'.

From Host A, the `symcg list` command with the `-ppath` option displays those composite groups that currently exist in the PowerPath database on Host A. Only the newly-created 3-sym-gns group currently exists. GNS has propagated this composite definition to Host B (as shown later).

```
symcg list -ppath
```

C O M P O S I T E G R O U P S							
Name	Type	Valid	Number of		Number of		
			Symms	RAGs	Devs	BCVs	VDEVs
3-sym-gns	RDF1	Yes	2	2	27	0	0

The `symcg show` command displays details of the composite group. Notice that the composite group (CG) is in PowerPath and in GNS as indicated by the "Yes" in those rows.

```
symcg show 3-sym-gns
```

```
Composite Group Name: 3-sym-gns
```

```
Composite Group Type      : RDF1
Valid                    : Yes
CG in PowerPath          : Yes
CG in GNS                 : Yes

Number of RDF (RA) Groups : 2
Number of STD Devices     : 27
Number of BCV's (Locally-associated) : 0
```

```

Number of VDEV's (Locally-associated)      : 0
Number of RBCV's (Remotely-associated STD-RDF) : 0
Number of BRBCV's (Remotely-associated BCV-RDF) : 0
Number of RRBCV's (Remotely-associated RBCV) : 0

```

```

Number of Symmetrix Units (2):
{

```

```

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

```

```

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

```

```

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

```

```

STD Devices (22):
{

```

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
/dev/vx/rdmp/c3t3d0s2	0110	RDF1	Enabled	12946
/dev/vx/rdmp/c3t3d1s2	0113	RDF1	Enabled	8631
/dev/vx/rdmp/c3t3d2s2	0115	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d3s2	0116	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d4s2	0117	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d5s2	0118	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d6s2	0119	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d7s2	011A	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d0s2	011B	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d1s2	011C	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d2s2	011D	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d3s2	011E	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d4s2	011F	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d5s2	0120	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d6s2	0121	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d7s2	0122	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d0s2	0123	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d1s2	0124	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d2s2	0125	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d3s2	0126	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d4s2	0127	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d5s2	0128	RDF1	Enabled	4315

```

}

```

```

2) Symmetrix ID      : 000184500160
   Microcode Version : 5568
   Number of STD Devices : 5
   Number of BCV's (Locally-associated) : 0
   Number of VDEV's (Locally-associated) : 0
   Number of RBCV's (Remotely-associated STD_RDF) : 0
   Number of BRBCV's (Remotely-associated BCV-RDF) : 0
   Number of RRBCV's (Remotely-associated RBCV) : 0

```

```

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

```

```

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

```

```
STD Devices (5):
```

```

{
  -----
  PdevName          Sym  Device  Consistency  Cap
                   Dev  Config  State         (MB)
  -----
  /dev/vx/rdmp/c5t2d24s2 0784 RDF1      Enabled      4315
  /dev/vx/rdmp/c5t2d25s2 0785 RDF1      Enabled      4315
  /dev/vx/rdmp/c5t2d26s2 0786 RDF1      Enabled      4315
  /dev/vx/rdmp/c5t2d27s2 0787 RDF1      Enabled      4315
  /dev/vx/rdmp/c5t2d28s2 0788 RDF1      Enabled      4315
  }
}

```

Now from Host B, issuing the `symcg list` command with the `-ppath` option displays the composite group that currently exists in the PowerPath database on Host B. The display reveals the same composite group that exists on Host A.

```
symcg list -ppath
```

```

          C O M P O S I T E      G R O U P S

Name          Type  Valid  Number of  Number of
                   Symms  RAGs    Devs    BCVs    VDEVs
3-sym-gns     RDF1  Yes    2         2       27      0      0

```

Also from Host B, the `symcg addall` command now adds to this composite group a range of devices from the third Symmetrix array (SID 35).

```
symcg -cg 3-sym-gns addall dev -range 150:168 -sid 35
```

Returning to Host A and issuing the `symcg list` command there displays whether the changes made to the composite group definition from Host B have been propagated to Host A. (You may have to wait up to 60 seconds to see the changes propagated.) The following display shows the addition of the third Symmetrix array and its devices to the composite group.

```
symcg list -ppath
```

```

          C O M P O S I T E      G R O U P S

Name          Type  Valid  Number of  Number of
                   Symms  RAGs    Devs    BCVs    VDEVs
3-sym-gns     RDF1  Yes    3         3       49      0      0

```

The `symcg show` command displays the updated details of the composite group. Note that the newly added devices from Symmetrix 000187900035 are not yet enabled for RDF consistency protection.

```
symcg show 3-sym-gns
```

```
Composite Group Name: 3-sym-gns
```

```

Composite Group Type      : RDF1
Valid                     : Yes
CG in PowerPath           : Yes
CG in GNS                 : Yes

```

```

Number of RDF (RA) Groups          : 3
Number of STD Devices              : 49
Number of BCV's (Locally-associated) : 0
Number of VDEV's (Locally-associated) : 0
Number of RBCV's (Remotely-associated STD-RDF) : 0
Number of BRBCV's (Remotely-associated BCV-RDF) : 0
Number of RRBCV's (Remotely-associated RBCV) : 0

```

Number of Symmetrix Units (3):

```

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

```

Number of RDF (RA) Groups (1):

```

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

```

STD Devices (22):

```

{
  -----
  PdevName          Sym  Device  Consistency  Cap
                   Dev   Config   State         (MB)
  -----
  /dev/vx/rdmp/c6t0d24s2  0150 RDF1+Mir   Disabled    12946
  /dev/vx/rdmp/c6t0d25s2  0153 RDF1+Mir   Disabled     8631
  /dev/vx/rdmp/c6t0d26s2  0155 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d27s2  0156 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d28s2  0157 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d29s2  0158 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d30s2  0159 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d31s2  015A RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d32s2  015B RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d33s2  015C RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d34s2  015D RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d35s2  015E RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d36s2  015F RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d37s2  0160 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d38s2  0161 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d39s2  0162 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d40s2  0163 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d41s2  0164 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d42s2  0165 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d43s2  0166 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d44s2  0167 RDF1+Mir   Disabled     4315
  /dev/vx/rdmp/c6t0d45s2  0168 RDF1+Mir   Disabled     4315
  }
}

```

```

2) Symmetrix ID          : 000184500160
   Microcode Version      : 5568
   Number of STD Devices  : 5
   Number of BCV's (Locally-associated) : 0
   Number of VDEV's (Locally-associated) : 0
   Number of RBCV's (Remotely-associated STD_RDF) : 0
   Number of BRBCV's (Remotely-associated BCV-RDF) : 0

```

Number of RRBCV's (Remotely-associated RBCV) : 0

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

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

STD Devices (5):
{

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
/dev/vx/rdmp/c5t2d24s2	0784	RDF1	Enabled	4315
/dev/vx/rdmp/c5t2d25s2	0785	RDF1	Enabled	4315
/dev/vx/rdmp/c5t2d26s2	0786	RDF1	Enabled	4315
/dev/vx/rdmp/c5t2d27s2	0787	RDF1	Enabled	4315
/dev/vx/rdmp/c5t2d28s2	0788	RDF1	Enabled	4315

}

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

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

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

STD Devices (22):
{

PdevName	Sym Dev	Device Config	Consistency State	Cap (MB)
/dev/vx/rdmp/c3t3d0s2	0110	RDF1	Enabled	12946
/dev/vx/rdmp/c3t3d1s2	0113	RDF1	Enabled	8631
/dev/vx/rdmp/c3t3d2s2	0115	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d3s2	0116	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d4s2	0117	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d5s2	0118	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d6s2	0119	RDF1	Enabled	4315
/dev/vx/rdmp/c3t3d7s2	011A	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d0s2	011B	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d1s2	011C	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d2s2	011D	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d3s2	011E	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d4s2	011F	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d5s2	0120	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d6s2	0121	RDF1	Enabled	4315
/dev/vx/rdmp/c3t4d7s2	0122	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d0s2	0123	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d1s2	0124	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d2s2	0125	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d3s2	0126	RDF1	Enabled	4315
/dev/vx/rdmp/c3t5d4s2	0127	RDF1	Enabled	4315

```

                                /dev/vx/rdmp/c3t5d5s2    0128 RDF1      Enabled      4315
                                }
                                }
    }

```

Another `symcg enable` command enables the remaining devices in the composite group for RDF consistency protection.

```
symcg -cg 3-sym-gns enable -noprompt
```

```
A consistency 'Enable' operation execution is in progress for composite group
'3-sym-gns'. Please wait...
```

```
The consistency 'Enable' operation successfully executed for composite group
'3-sym-gns'.
```

This chapter provides examples of the Symmetrix command line interface (SYMCLI) actions and specific commands, which are used to measure and track changes to data on a group of Symmetrix volumes.

- ◆ [Change Tracker example](#)16-2

Change Tracker example

The test setup consists of ten 4-GB standard devices (Symmetrix device names 108 to 111) and ten 4-GB R1-BCV devices (Symmetrix device names 112 to 11B) on a local Symmetrix unit. The R1-BCV devices are first established with the standard devices as local BCV pairs, thereby suspending their RDF links. The R1-BCVs are then split from their standard devices and resynchronized with their R2 mirrors on a remote Symmetrix unit. Subsequently, the BCVs are re-established with their standard devices, and changes to the local BCV pairs are monitored with Change Tracker from that point in time.

Creating a device group and adding devices to it are prerequisites for establishing BCV pairs. The `syndg` command creates a device group called `deltadg`. The `symlid` command adds standard devices to this device group, and the `symbcv` command associates BCVs with the device group. The `-range` option is helpful in specifying a large number of standards and BCVs in one command line.

```
syndg create deltdag
symlid -g deltdag -range 108:111 addall dev
symbcv -g deltdag -range 112:11B associateall dev
```

The `symlid list` command shows the device group details.

```
symlid -g deltdag list
```

```
Device Group (DG) Name: deltdag
DG's Type               : REGULAR
DG's Symmetrix ID       : 000184600266
```

Standard Device Name			Directors			Device		
Logical	Physical	Sym	SA	:P DA :IT	Config	Att	Sts	Cap (MB)
DEV001	c3t13d9s2	108	14A:1	02B:D3	2-Way Mir	RW		4096
DEV002	c3t13d10s2	109	14A:1	01A:C3	2-Way Mir	RW		4096
DEV003	c3t13d11s2	10A	14A:1	01B:D2	2-Way Mir	RW		4096
DEV004	c3t13d12s2	10B	14A:1	02A:C2	2-Way Mir	RW		4096
DEV005	c3t13d13s2	10C	14A:1	02B:D1	2-Way Mir	RW		4096
DEV006	c3t13d14s2	10D	14A:1	01B:D1	2-Way Mir	RW		4096
DEV007	c3t13d15s2	10E	14A:1	02A:D1	2-Way Mir	RW		4096
DEV008	c3t13d16s2	10F	14A:1	01A:D1	2-Way Mir	RW		4096
DEV009	c3t13d17s2	110	14A:1	02B:C1	2-Way Mir	RW		4096
DEV010	c3t13d18s2	111	14A:1	01B:C1	2-Way Mir	RW		4096

BCV Devices associated with this dg:

BCV Device				Standard Device				Status
Logical	Sym	RDF	Att.	Inv. Tracks	Logical	Sym	Inv. Tracks	BCV <=> STD
BCV001	112	R1		0 N/A		N/A	0	NeverEstab
BCV002	113	R1		0 N/A		N/A	0	NeverEstab
BCV003	114	R1		0 N/A		N/A	0	NeverEstab
BCV004	115	R1		0 N/A		N/A	0	NeverEstab
BCV005	116	R1		0 N/A		N/A	0	NeverEstab
BCV006	117	R1		0 N/A		N/A	0	NeverEstab
BCV007	118	R1		0 N/A		N/A	0	NeverEstab
BCV008	119	R1		0 N/A		N/A	0	NeverEstab
BCV009	11A	R1		0 N/A		N/A	0	NeverEstab
BCV010	11B	R1		0 N/A		N/A	0	NeverEstab

```

Total          -----
Track(s)              0              0
MB(s)                0.0            0.0

```

Legend for BCV devices:

(+): BCV is paired with a member STD device.

(-): BCV is paired with a non-member STD device.

First, the local BCV pairs are established. A full synchronization is required because this is the first establish of the BCV pairs.

symmir -g deltadg -full -exact -noprompt establish

```

'Full Establish' operation execution is in progress
  for device group 'deltadg'.
Please wait...
'Full Establish' operation successfully initiated
  for device group 'deltadg'.

```

The symrdf query command with the -bcv option shows that the RDF links between the R1-BCVs and their R2s have been suspended as a result of establishing the local BCV pairs. The links are NR (Not Ready).

symrdf -g deltadg -bcv query

```

Device Group (DG) Name: deltadg
DG's Type              : REGULAR
DG's Symmetrix ID      : 000184600266

```

```

                                R E M O T E      S Y M M E T R I X

```

Source (R1) View					Target (R2) View					M O D E S		
BCV	ST			LI	ST			M				
Logical	A			N	A			O				
Device	T R1 Inv	R2 Inv	K	T R1 Inv	R2 Inv	d		RDF Pair				
Dev	E Tracks	Tracks	S	Dev	E Tracks	Tracks	e	Dom ACp	STATE			
BCV001	112 NR	0	0 NR	135 WD	0	0	0 SYN DIS OFF	Suspended				
BCV002	113 NR	0	0 NR	136 WD	0	0	0 SYN DIS OFF	Suspended				
BCV003	114 NR	0	0 NR	137 WD	0	0	0 SYN DIS OFF	Suspended				
BCV004	115 NR	0	0 NR	138 WD	0	0	0 SYN DIS OFF	Suspended				
BCV005	116 NR	0	0 NR	139 WD	0	0	0 SYN DIS OFF	Suspended				
BCV006	117 NR	0	0 NR	13A WD	0	0	0 SYN DIS OFF	Suspended				
BCV007	118 NR	0	0 NR	13B WD	0	0	0 SYN DIS OFF	Suspended				
BCV008	119 NR	0	0 NR	13C WD	0	0	0 SYN DIS OFF	Suspended				
BCV009	11A NR	0	0 NR	13D WD	0	0	0 SYN DIS OFF	Suspended				
BCV010	11B NR	0	0 NR	13E WD	0	0	0 SYN DIS OFF	Suspended				

```

Total          -----
Track(s)              0              0
MB(s)                0.0            0.0

```

The symmir verify command is used to check on the progress of the BCV pair establish operation and confirm when it has completed.

symmir -g deltadg verify -i 60

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

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

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

A `symmir query` command confirms that the BCV pairs have been established and synchronization is complete.

`symmir -g deltadg query`

```
Device Group (DG) Name: deltadg
DG's Type           : REGULAR
DG's Symmetrix ID   : 000184600266
```

Standard Device			BCV Device			State
Logical	Inv.		Logical	Sym	Inv.	
	Sym	Tracks			Tracks	STD <=> BCV
DEV001	108	0	BCV001	112 *	0	Synchronized
DEV002	109	0	BCV002	113 *	0	Synchronized
DEV003	10A	0	BCV003	114 *	0	Synchronized
DEV004	10B	0	BCV004	115 *	0	Synchronized
DEV005	10C	0	BCV005	116 *	0	Synchronized
DEV006	10D	0	BCV006	117 *	0	Synchronized
DEV007	10E	0	BCV007	118 *	0	Synchronized
DEV008	10F	0	BCV008	119 *	0	Synchronized
DEV009	110	0	BCV009	11A *	0	Synchronized
DEV010	111	0	BCV010	11B *	0	Synchronized
Total	-----				-----	
Track(s)		0				0
MB(s)		0.0				0.0

Legend:

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

The `-remote` option with `symmir split` command has the effect of automatically bringing up the RDF connection between the R1-BCVs and their R2 mirrors upon completion of the split. Without the `-remote` option, the `symrdf -g deltadg establish -bcv` command would have to follow the `symmir -g deltadg -noprompt split` command to establish the BCV-R1s and their remote mirrors.

`symmir -g deltadg split -noprompt -remote`

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

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

The `symrdf verify` command is used to check on the progress of the SRDF establish operation and confirm when it has completed.

`symrdf -g deltadg verify -i 30`

```
Not all devices in the RDF group 'deltadg' are in the 'Synchronized' state.
Not all devices in the RDF group 'deltadg' are in the 'Synchronized' state.
All devices in the RDF group 'deltadg' are in the 'Synchronized' state.
```

A `symrdf` query command confirms that the SRDF pairs have been established and synchronization is complete.

`symrdf -g deltadg -bcv query`

```
Device Group (DG) Name: deltadg
DG's Type           : REGULAR
DG's Symmetrix ID   : 000184600266
```

R E M O T E S Y M M E T R I X									
Source (R1) View					Target (R2) View				M O D E S
-----					-----				-----
		ST		LI	ST		M		
BCV		A		N	A		o		
Logical	T R1 Inv	R2 Inv	K	T R1 Inv	R2 Inv	d			RDF Pair
Device	Dev	E Tracks	Tracks	S Dev	E Tracks	Tracks	e	Dom ACp	STATE

BCV001	112 RW	0	0 RW	135 WD	0	0	SYN DIS OFF		Synchronized
BCV002	113 RW	0	0 RW	136 WD	0	0	SYN DIS OFF		Synchronized
BCV003	114 RW	0	0 RW	137 WD	0	0	SYN DIS OFF		Synchronized
BCV004	115 RW	0	0 RW	138 WD	0	0	SYN DIS OFF		Synchronized
BCV005	116 RW	0	0 RW	139 WD	0	0	SYN DIS OFF		Synchronized
BCV006	117 RW	0	0 RW	13A WD	0	0	SYN DIS OFF		Synchronized
BCV007	118 RW	0	0 RW	13B WD	0	0	SYN DIS OFF		Synchronized
BCV008	119 RW	0	0 RW	13C WD	0	0	SYN DIS OFF		Synchronized
BCV009	11A RW	0	0 RW	13D WD	0	0	SYN DIS OFF		Synchronized
BCV010	11B RW	0	0 RW	13E WD	0	0	SYN DIS OFF		Synchronized
Total									
Track(s)		0	0	0		0	0		
MB(s)		0.0	0.0	0.0		0.0	0.0		

At this point, the BCV pairs are re-established using a `symmir establish` command. This action suspends the RDF links between the R1-BCVs and their R2 target devices.

`symmir -g deltadg -noprompt establish`

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

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

The resynchronization is fast here because nothing has changed since the BCVs and standards were last established.

A `symmir verify` command is used to check on the BCV pair state.

`symmir -g deltadg verify`

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

To monitor the volume changes, create a DeltaMark session for each standard device in the device group.

`symchg -g deltadg create`

```
10 Change Tracker DeltaMark sessions have been created
for Device Group deltadg.
```

A listing of the sessions on the local Symmetrix system shows the 10 sessions (one for each standard device in the device group). The `symchg list` command lists *all* current DeltaMark sessions associated with objects in the specified Symmetrix array (sid 66).

```
symchg -sid 66 -session list
```

```
Symmetrix ID: 000184600266
```

C H A N G E T R A C K E R D E L T A M A R K S E S S I O N S									
Device Name		Directors				Device			

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

/dev/rdsk/c3t13d9s2	108	14A:1	02B:D3	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d10s2	109	14A:1	01A:C3	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d11s2	10A	14A:1	01B:D2	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d12s2	10B	14A:1	02A:C2	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d13s2	10C	14A:1	02B:D1	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d14s2	10D	14A:1	01B:D1	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d15s2	10E	14A:1	02A:D1	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d16s2	10F	14A:1	01A:D1	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d17s2	110	14A:1	02B:C1	2-Way	Mir	Grp'd	RW	4096	
/dev/rdsk/c3t13d18s2	111	14A:1	01B:C1	2-Way	Mir	Grp'd	RW	4096	

```
10 Change Tracker DeltaMark sessions were found.
```

Using the marking action updates the Change Tracker database and resets the bitmap for each DeltaMark session. This is the starting point for volume change measurement.

```
symchg -g deltadg mark
```

```
The Device Group deltadg has successfully been marked
on Wed Nov 21 2001 15:46:14
```

The `symchg view` command writes to a log file, called `deltalog`, that is used later to generate Change Tracker reports and perform capacity-planning calculations. Change information entries are appended to the log file if it already exists; the log file is created if it does not already exist (as in this case). Change information entries are made in the log file every two minutes (120 seconds), and logging is halted after 30 entries are made in the log file.

```
symchg -g deltadg -log ./deltalog -i 120 -c 30 view
```

In a separate terminal session, a C shell script, called `changetracks`, is executed to create changes on the standard volumes. This script writes 3 GB of data on each of the 10 standard volumes. Since a track of data is 32 KB, every 32 tracks constitute 1 MB, and 32,000 tracks constitute a GB.

```
cat changetracks
```

```
#!/bin/csh
echo Starting writes on device group deltadg at `date`
foreach i (9 10 11 12 13 14 15 16 17 18)
set disk = /dev/rdisk/c3t13d{ $i }s2
dd if=/dev/zero of=$disk bs=32768 count=96000
end
echo Ending writes on device group deltadg at `date`
```

```
changetracks
```

```
Starting writes on device group deltadg at Wed Nov 21 15:47:22 EST 2001
96000+0 records in
96000+0 records out
```

```

96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out
96000+0 records in
96000+0 records out

```

Ending writes on device group deltadg at Wed Nov 21 16:23:35 EST 2001

The deltalog log file contains the following entries. The entries after 16:25 are repetitive and were truncated from the file.

cat deltalog

```

11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 0 , 131070 , SUM
11/21/2001 15:46:20 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 59525 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 0 , 131070 , SUM
11/21/2001 15:48:24 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 91946 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 0 , 131070 , SUM
11/21/2001 15:50:27 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 96000 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 77643 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 0 , 131070 , SUM
11/21/2001 15:52:30 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 0 , 131070 , SUM

```

[illegible]

[illegible]

```

11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 96000 , 131070 , SUM
11/21/2001 16:21:16 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 54038 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 96000 , 131070 , SUM
11/21/2001 16:23:19 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 91920 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 108 , /dev/rdisk/c3t13d9s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 109 , /dev/rdisk/c3t13d10s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10A , /dev/rdisk/c3t13d11s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10B , /dev/rdisk/c3t13d12s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10C , /dev/rdisk/c3t13d13s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10D , /dev/rdisk/c3t13d14s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10E , /dev/rdisk/c3t13d15s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 10F , /dev/rdisk/c3t13d16s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 110 , /dev/rdisk/c3t13d17s2 , 96000 , 131070 , SUM
11/21/2001 16:25:23 , 000184600266 , DG , deltadg , 111 , /dev/rdisk/c3t13d18s2 , 96000 , 131070 , SUM

```

The `deltalog` log file can be used to generate "what if" scenarios with Change Tracker. In the following examples, an RDF (RA) link should be thought of as a communications channel. While a Symmetrix array may be configured with two remote link directors, those directors are often connected through a channel converter to one T3 link when communication to a target Symmetrix array is over an extended distance. Therefore, to calculate the synchronization time in this case, it is necessary to specify a start time, a stop time, a single RA link (communication channel), and the expected channel transfer rate.

The following calculation shows that the RDF synchronization of these 10 R1-BCV devices with their R2 devices would require 1 hour, 53 minutes if a 4.5 MB/s link is used. (This is the approximate bandwidth of a T3 link.) Of course this does not take into account the effects of factors like link utilization, compression, and so on. However, Change Tracker does provide an accurate picture of the changed data, as evidenced by the fact that the number of changed tracks reported by Change Tracker coincides with the number of tracks written by the `dd` command in the `changetracks` script (see the previous `cat changetracks` command).

```

symchg -g deltadg -ra 1 -log deltalog -start 11212001 15:46
      -stop 11212001 16:26 report -rate 4500

```

Device group : deltadg

```

Start time   : Wed Nov 21 2001 15:46:20
Stop time    : Wed Nov 21 2001 16:25:23

```

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	960000	73	1	4500	01:53

With the same start and stop times and transfer rate, using two and four RDF links (communication channels) produces synchronization times that are proportionately less.

```
symchg -g deltadg -ra 2 -log deltalogs -start 11212001 15:46  
-stop 11212001 16:26 report -rate 4500
```

Device group : deltadg

Start time : Wed Nov 21 2001 15:46:20
Stop time : Wed Nov 21 2001 16:25:23

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	960000	73	2	4500	00:56

```
symchg -g deltadg -ra 4 -log deltalogs -start 11212001 15:46  
-stop 11212001 16:26 report -rate 4500
```

Device group : deltadg

Start time : Wed Nov 21 2001 15:46:20
Stop time : Wed Nov 21 2001 16:25:23

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	960000	73	4	4500	00:28

Change Tracker can also be used to calculate other variables. For example, the following command shows how to calculate the link bandwidth if the resynchronization time is specified. A resynchronization time of 113 minutes is used to enable a comparison between this result and a previous calculation that used a single RDF link (communication channel).

```
symchg -g deltadg -ra 1 -log deltalogs -start 11212001 15:46  
-stop 11212001 16:26 report -resync 113
```

Device group : deltadg

Start time : Wed Nov 21 2001 15:46:20
Stop time : Wed Nov 21 2001 16:25:23

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	960000	73	1	4530	01:53

The following command shows how to calculate the number of RDF links (communication channels) needed if the bandwidth and resynchronization time are specified.

```
symchg -g deltadg -log deltalogs -start 11212001 15:46 -stop 11212001 16:26  
-resync 113 -rate 4500 report
```

Device group : deltadg

Start time : Wed Nov 21 2001 15:46:20
Stop time : Wed Nov 21 2001 16:25:23

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	960000	73	1	4500	01:53

The following command calculates the bandwidth needed to transfer changes every 20 minutes, using two RDF links (communication channels) and allowing 10 minutes for the transfer. This calculation shows that it is possible to transfer information at 20-minute intervals with two 14.7 MB/s links. Ten of those 20 minutes could be set aside for SRDF resynchronization (transfer), and the other ten could be used to reestablish the BCV pairs, and then split off the BCVs. Observe that the number of changed tracks is smaller than in the previous examples since only 20 minutes worth of changes were used from the log file (instead of the full 40 minutes used in the earlier examples).

```
symchg -g deltadg -ra 2 -log deltalog -start 11212001 15:46 -stop 11212001 16:07 -resync 10 report
```

Device group : deltadg

Start time : Wed Nov 21 2001 15:46:20

Stop time : Wed Nov 21 2001 16:06:53

SymmID	Num of Devices	Total Tracks	Tracks Changed	%Tracks Changed	Num of RA's	BndWth (KBS)	Sync Time
000184600266	10	1310700	549942	41	2	14665	00:10

A

- abort locks 4-19
- access control commands
 - introduction 4-2
- access control database 4-3
- access control entries
 - see also* ACEs
- access control lists
 - see also* ACLs
- access control strategy for
 - TimeFinder/BCV 4-22
 - TimeFinder/Snap 4-22
- access groups 4-3
 - access IDs (PINs) 4-8
 - adding access IDs 4-7
 - creating 4-7
 - deleting 4-9
 - editing/managing 4-8
 - granting permissions 4-13
 - listing 4-18
 - moving access IDs 4-8
 - removing access IDs 4-8
 - removing ACEs 4-9
 - removing permissions 4-17
 - setting up 4-7
 - showing 4-18
 - unknown 4-21
- access IDs 4-3, 4-7
 - viewing 4-18
- access modes
 - SYMCLI commands 2-8
- access pools 4-2, 4-10
 - add devices 4-11
 - creating 4-10
 - deleting 4-12
 - editing/managing 4-11
 - listing 4-18
 - remove device 4-11
 - setting up 4-10
 - showing 4-18
- access types 4-13
- ACEs
 - creating 4-13

- listing 4-18
- showing 4-18

ACLs

- listing 4-18
- ADMIN permissions 4-10
- ADMIN privileged
 - establishing 4-21
- ADMIN rights 4-13
- applications registered 3-9
- array authorization 2-5
- audit log
 - filtering log entries 11-6
 - symaudit 11-6
- authentication types 5-4
- authority
 - ADMIN permissions 4-10
 - ADMIN rights 4-13
- authorization 2-5

B

- backup database 4-23
- BCV devices 3-23
- Business continuance operations 3-23
- Business Continuance Volume 3-23

C

- cache memory 1-4
 - LRU defined 1-4
- cache statistics
 - LRU 8-16
- CLARiiON array
 - discover 2-3, 2-6
- command file
 - access control operations 4-5
 - commit 4-5, 5-7
 - possible operations 4-5
 - prepare 4-5
 - preview 4-5, 5-7
- commit command file 4-5, 5-7
- common audit log 11-6
- configuration database, *see also* database
- creating access pools 4-10

D

- DA 1-3
 - prefetch statistics 8-9
- database
 - access control 4-3
 - access modes 2-7
 - backup 4-23
 - building a Symmetrix 2-2
 - changing file in use 2-7
 - configuration information 2-9
 - inhibiting synchronization 2-8
 - location 2-7
 - multi-database environment 2-7
 - offline 2-7
 - online 2-7
 - restore 4-23
 - synchronization 2-8
- DELTA reports 9-9
- DeltaMark bitmap 9-2
- device configuration 3-32
- device groups
 - adding all ungrouped devices 6-8
 - adding devices to 6-5, 6-6
 - BCV devices in 6-3
 - creating 6-5
 - deleting 6-15
 - exporting 6-11
 - gatekeepers associated with 6-4
 - information about 6-9
 - listing 6-9
 - moving a device from 6-13
 - naming 6-3
 - RDF1 6-3, 6-5
 - RDF2 6-3, 6-5
 - REGULAR 6-3, 6-5
 - removing a device from 6-15
 - renaming 6-13
 - SRDF standard devices in 6-3
 - types 6-3
 - using 6-2
 - virtual devices in 6-3
- device lists
 - performing operation on 1-8
- device pools. *see also* access pools
- device service states 3-28
- device status 3-28
- device storage objects 9-5
- device text file 9-5
- devices
 - renaming 6-13
- directors
 - configuration of 3-12
 - offline 3-14
 - online 3-14
 - statistics 8-10, 8-12
 - status of 3-12
- discover
 - base daemon usage 2-2
 - CLARiiON 2-3, 2-6
 - Symmetrix 2-3, 2-6

- disk

- statistics 8-3

- disk director 1-3

- disks

- by disk group 3-35
 - configuration of 3-34
 - failed disk details 3-36
 - hostpares 3-36

- DMSP 8-14

- dynamic mirroring service policy 8-14

- dynamic RDF 3-23

E

- enforcement 5-5

- export groups to files 6-11

- External device locks 3-32

F

- files

- device import list 1-8

- event logs 11-2

- front-end director 1-3

G

- gatekeeper devices 3-23

- associating 10-6

- candidate priority 10-2

- choosing 10-3

- database considerations of 10-4

- definition 10-2, 10-3

- disassociating 10-6

- establishing 10-5

- obtain list of 10-6

- sizing 10-4

- symgate 10-5 to 10-6

- undefining 10-5

- GNS

- active group list 7-6

- backup, automatic 7-17

- backup, manual 7-17

- composite groups 7-6

- daemon 7-2, 7-10, 7-11

- device groups 7-2

- output 7-16

- PowerPath interaction 7-5

- restore, manual 7-17

- shared repository 7-2

- SRDF option 7-5

- GNS, daemon options

- Remote mirrors 7-13

- Remote mirrors, no BCV 7-13

- granting permissions 4-13

- Group Name Services, *see also* GNS

- groups 4-7

H

- HBA details 3-3

- host database file, *see also* database

hosts connected 3-10
 hyper 1-4, 3-22
 hyper-volume 1-4, 3-22

L

locks
 external device 3-32
 locked sessions 4-19
 releasing locked session 4-19
 Symmetrix 3-15, 3-16
 logical device name 6-3
 logs 11-2
 configuration format 11-3
 dated and undated file names 11-3
 enabling or disabling 11-2
 log entry date formats 11-3
 naming 11-2
 LRU
 cache management 8-16
 defined 1-4

M

managing access groups 4-7
 managing access pools 4-10
 marked objects 9-6
 memory board information 3-18
 meta devices 3-23
 MSC 6-14, 6-19
 multi session consistency 6-19

N

network services 3-18

O

offline
 director ports 3-14
 directors 3-14
 online
 director ports 3-14
 directors 3-14

P

permissions 4-13
 removing permissions 4-17
 physical devices
 list configuration 3-32
 lists 3-24
 show configuration 3-32
 pools, access 4-10
 prepare command file 4-5
 preview command file 4-5, 5-7

R

RDF consistency 6-14
 regular groups 6-3, 6-5
 restore database 4-23
 Role 5-2

S

SAVE devices 3-24
 SCSI INQUIRY 3-3
 SEL, *see also* Symmetrix External Locks
 service states 3-28
 showing ACEs 4-18
 SNMP traps 11-8
 SRDF devices 3-23, 4-2
 SRDF strategy 4-22
 standard devices 3-23
 statistics
 BCV devices 8-13
 cache activity 8-3, 8-22
 cache memory to disk 8-3
 director 8-10
 director port 8-12
 I/O requests 8-3
 LRU 8-16
 performance 8-2
 prefetch activity for DAs 8-9
 prefetch disk activity 8-3
 standard devices 8-13
 SYMCLI commands for 8-4
 status of device 3-28
 storage capacities 3-11
 storgnsd, *see also* GNS
 storgnsd, *see also* GNS, daemon options
 strategies
 access control 4-20
 SUM reports 9-9
 SYMAPI database, *see also* database
 symauth 5-2, 5-5
 SYMCLI commands
 -output xml option 12-2
 symacl 4-5 to 4-23
 symaudit 11-6 to 11-7
 symauth 5-2, 5-5, 5-7
 symcfg 2-2 to 2-9, 3-7 to 3-21
 symcgr 6-18 to 6-27, 7-6
 symchg 9-3 to 9-14
 symcli 1-5, 2-7
 symdev 3-24 to 3-33
 symdg 6-5, 6-13, 7-6
 syndisk 3-34 to 3-36
 synevent 11-4 to 11-5
 symgate 10-5 to 10-6
 syminq 3-3 to 3-6
 symld 6-6 to 6-14
 sympd 3-24
 symstat 8-4 to 8-23
 Symmetrix
 device configuration 3-32
 Symmetrix array
 discover 2-3, 2-6
 Symmetrix devices
 list 3-25
 Symmetrix External Locks
 details 3-16
 releasing 3-16
 viewing 3-15

U

- unknown access groups 4-21
- user access IDs, PINs 4-8
- user names 5-4
 - current user 5-4

V

- VCM devices 3-24
- viewing access controls 4-18
- virtual devices 3-23

X

- XML output 12-2