

**EMC Solutions Enabler
Symmetrix Base Management CLI**
Version 6.0

PRODUCT GUIDE
P/N 300-000-874
REV A06

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

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

Printed January, 2005

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

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

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

Trademark Information

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

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

Chapter 1 Solutions Enabler Overview

SYMCLI Introduction	1-2
Symmetrix Component Overview	1-3
Directors in the Enginuity Environment.....	1-3
Cache	1-4
Storage Devices	1-4

Chapter 2 Solutions Enabler Environment

Command Line Environment Setup	2-2
Set SYMCLI Path.....	2-2
Starting and Stopping the Server	2-3
Environment Variables.....	2-3
Options File for Optional Behavior Parameters.....	2-4
Examining SYMCLI Versions.....	2-4
Avoidance and Selection Files.....	2-5
symavoid.....	2-6
inqfile.....	2-6
gkavoid.....	2-6
gkselect.....	2-7
Symmetrix Event Logs	2-8
How to Configure Log Options.....	2-9
Monitoring Symmetrix Events via SYMCLI.....	2-11
Common Audit Log	2-13
SYMCLI Usage Tips.....	2-16
Obtaining SYMCLI Help	2-16
Referencing Devices and Objects	2-17

Altering the Command Line Response	2-18
Presetting the Names and IDs	2-19
Time-Saving Command Input Tips.....	2-19
Truncated Output	2-20
Performing Operations on a Device List.....	2-20

Chapter 3 **Discovery**

Discovering Configuration and Status Data.....	3-2
Performing Discovery	3-2
Scanning the Hardware	3-3
Synchronizing Configuration Updates with the Database.....	3-4
Sync Operations.....	3-4
SYMAPI Configuration Database File	3-5
Database File Location.....	3-5
Changing the Current Database File	3-6
Access Modes	3-6
Command Modes: Online and Offline	3-6
Database Configuration Information	3-7

Chapter 4 **Configuration Data**

Levels of Configuration Data.....	4-2
SCSI-Level.....	4-2
Array-Level	4-2
Device-Level.....	4-2
Disk-Level.....	4-2
SCSI Inquiries.....	4-3
Scope of Inquiries	4-3
SCSI Inquiry Device Information: Examples.....	4-3
HBA Information.....	4-6
Array Data	4-7
List of Symmetrix Arrays	4-7
Symmetrix Configuration Data	4-8
Viewing Application Registrations with Symmetrix Access.....	4-9
Viewing Hosts with Symmetrix Access	4-10
Viewing Storage Capacities to Hosts.....	4-11
Viewing Network Services.....	4-11
Viewing Memory Board Information	4-11
Mainframe CU Image	4-12
Director Configuration Data	4-13
LRU Cache Management Group.....	4-16

Symmetrix External Locks.....	4-17
Taking RA Directors Offline	4-19
Taking Front-End Director Ports Offline	4-19
Symmetrix Devices	4-20
Device Emulation.....	4-20
Device Type Overview	4-21
List of Devices.....	4-22
Device Emulation Type	4-30
Filtering Device Data with sympd	4-31
Filtering Symmetrix Device Data	4-32
Detailed Device Configuration Data.....	4-34
Releasing Device External Locks	4-35
CLARiiON Devices.....	4-36
Example: CLARiiON Devices	4-36
Disk-Level Data	4-37
List and Show Disks	4-37
Disk Groups.....	4-38

Chapter 5 Gatekeeper Devices

Overview	5-2
Gatekeeper Candidates	5-2
How SYMCLI Uses Gatekeepers	5-3
Defined Gatekeepers	5-4
How to Size a Gatekeeper.....	5-4
Database Considerations	5-5
SYMCLI Gatekeeper Management.....	5-6
Defining a Gatekeeper.....	5-6
Undefining a Gatekeeper.....	5-7
Listing the Gatekeeper Devices.....	5-8
Associating with a Device Group	5-8
Disassociating from a Device Group.....	5-8

Chapter 6 Grouping Devices

Overview	6-2
Groupings	6-2
Group Name Services (GNS).....	6-3
Device Groups	6-4
Group Names and Device Names	6-4
Device Group Types	6-5
How to Create a Device Group.....	6-6
Adding Devices	6-7

Listing Device Groups	6-9
Show Details about a Device Group	6-10
Exporting and Importing Device Lists	6-14
Renaming Devices	6-16
Adding Devices from a Device Group to a Composite Group	6-16
Removing Devices	6-17
Moving Devices	6-18
Renaming a Device Group	6-19
Deleting a Device Group	6-19
Composite Groups	6-20
What Is an RDF Consistency Group?	6-21
Multi Session Consistency (MSC)	6-21
PowerPath Consistency	6-22
Creating Composite Groups	6-24
Moving a Composite Group to or from MSC or PowerPath	6-26
Adding Standard Devices	6-28
Removing Standard Devices	6-29
Renaming a Composite Group	6-30
Composite Group Creation and Output Example	6-30
Composite Group Lists	6-33
Changing the State of Devices in a Composite Group	6-34
Hold Devices in a Composite Group	6-35
Unhold Devices in a Composite Group	6-35
SYMAPI Optional Behavior Parameters for Composite Groups	6-36

Chapter 7 Group Name Services

Group Name Services Overview	7-2
How the GNS Daemon Works	7-2
Shared Group Definitions with GNS	7-3
How GNS Handles Updates to Group Definitions	7-5
GNS and Consistency Groups	7-6
GNS Device Groups and SRDF	7-6
GNS Behavior in Client/Server Mode	7-7
Active vs. Inactive Group Lists	7-8
GNS Configuration	7-9
How to Set Up the GNS Option	7-9
How to Configure GNS in a Multi-Host Environment	7-9
GNS User Authentication	7-10
How to Configure GNS for Consistency	7-10

GNS Daemon Control.....	7-11
Enabling the Watchdog	7-12
Managing the GNS Daemon	7-12
GNS Daemon and Symmetrix External Locks.....	7-12
GNS Daemon Log File.....	7-13
GNS Daemon Options File	7-13
How to Mirror Remote Device Group Definitions.....	7-17
Controlling the Mirrored Group.....	7-17
Impact of Remotely Mirroring Group Definitions	7-18
BCV and Remote BCV Device Alias Name Swap	7-18
Composite Group Logical Device Name Support	7-20
Identify GNS Groups.....	7-21
Group Output.....	7-21
GNS Backup and Restore Mechanism	7-23
Automatic GNS Backups	7-23
Manual Backups and Restores via storgnsd	7-24
Toubleshooting GNS.....	7-25
Group Name Collisions.....	7-25
Inconsistent Groups.....	7-26

Chapter 8 **Statistics**

Overview	8-2
About the symstat Command	8-2
Reporting Objects.....	8-3
Performance Type Options	8-3
REQUEST Statistics.....	8-5
Array Statistics	8-5
Device Statistics.....	8-6
SRDF/ A Session Statistics	8-7
Example: All Device Groups	8-8
BACKEND Statistics and PREFETCH Activity	8-10
Returning BACKEND Statistics.....	8-10
PREFETCH Activity	8-10
Director Statistics.....	8-12
Director I/O Requests	8-12
PORT Statistics.....	8-14
How to Return Port Statistics.....	8-14
MEMIO Statistics.....	8-15
Example: Memory and I/O Statistics.....	8-15
DMSP Stats.....	8-17
Device Dynamic Mirroring Service Policy Statistics.....	8-17
Disk Statistics	8-18

Specifying a Disk	8-18
CACHE Statistics	8-19
Front-End Director Statistics	8-19
LRU Cache Management Statistics	8-19
SRDF/A Session Statistics	8-21
CYCLE Statistics	8-25
Example: All Device Groups	8-26
 Appendix A XML Structured Output	
XML Structured Output Data	A-2
XSLT: XML Data Transformations	A-2
XML Output Options	A-3
 Index	i-1

Figures

1-1	SYMCLI Host/Array Interaction	1-2
1-2	Symmetrix Components	1-3
6-1	RDF Consistency Groups	6-22
7-1	GNS Daemon Controls GNS Activity	7-2
7-2	Host-Visible GNS State	7-4
7-3	GNS Across Multiple Hosts	7-5
7-4	Mirrored Remote Group Definition	7-19

Tables

2-1	Log File Configuration Options	2-9
2-2	SYMCLI Help Options	2-16
2-3	Command Naming Conventions	2-17
3-1	Database Synchronization Options	3-4
4-1	Device Types	4-21
7-1	GNS Daemon Optional Behavior Parameters	7-14
8-1	Performance Display Options	8-3

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

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

Audience This manual is intended for command-line users and script programmers interested in details about using the Base Management 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, *Solutions Enabler Overview*, highlights the major Symmetrix features, the Base Management SYMCLI commands, and SYMCLI terminology.

Chapter 2, *Solutions Enabler Environment*, describes how to setup your command line environment, how to configure avoidance and selection files, and details about Symmetrix event logs and logging. In addition, this chapter provides some tips on SYMCLI usage.

Chapter 3, *Discovery*, describes the discovery process and how to retrieve Symmetrix array- and device-level configuration data. In addition, it details the Symmetrix configuration database.

Chapter 4, *Configuration Data*, describes how to use the SYMCLI to manage discovered Symmetrix array- and device-level configuration data.

Chapter 5, *Gatekeeper Devices*, describes how to use gatekeeper devices.

Chapter 6, *Grouping Devices*, describes how to manage device groups and composite groups.

Chapter 7, *Group Name Services*, describes the optional Group Name Services (GNS) feature.

Chapter 8, *Statistics*, describes how to display statistics for various Symmetrix devices.

Appendix A, *XML Structured Output*, describes the XML output option of the SYMCLI.

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

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 Version 6.0 Support Matrix*
- ◆ *EMC Host Connectivity Guides*

Conventions Used in This Manual

The following conventions are used in this manual:

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

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.

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



CAUTION

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

Typographical Conventions

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

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

Where to Get Help

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

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

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

Language services are available upon request.

Sales and Customer Service Contacts

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

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

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

<http://powerlink.emc.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please e-mail us at **techpub_comments@emc.com** to let us know about your opinion or any errors concerning this manual.

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

Solutions Enabler Overview

This chapter provides an overview of the Solutions Enabler Symmetrix Command Line Interface (SYMCLI) that focuses on the tasks of managing a Symmetrix storage environment. The chapter describes terminology and the interface between the host and the Symmetrix systems. It also overviews and identifies the basic Symmetrix operations and the SYMCLI Base Management command set.

- ◆ SYMCLI Introduction1-2
- ◆ Symmetrix Component Overview1-3

SYMCLI Introduction

The 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.

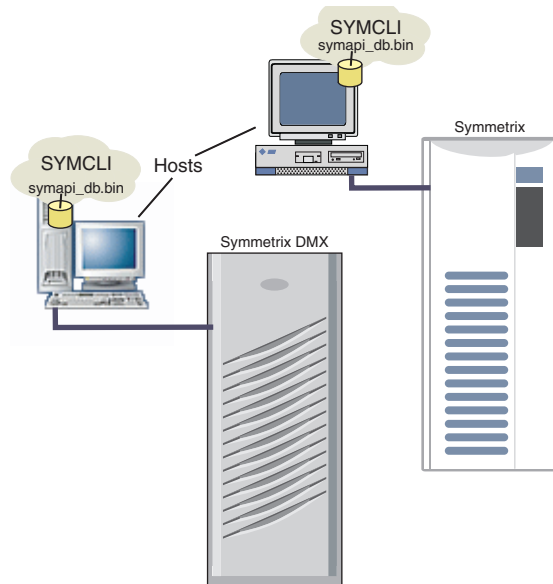


Figure 1-1 SYMCLI Host/Array Interaction

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.

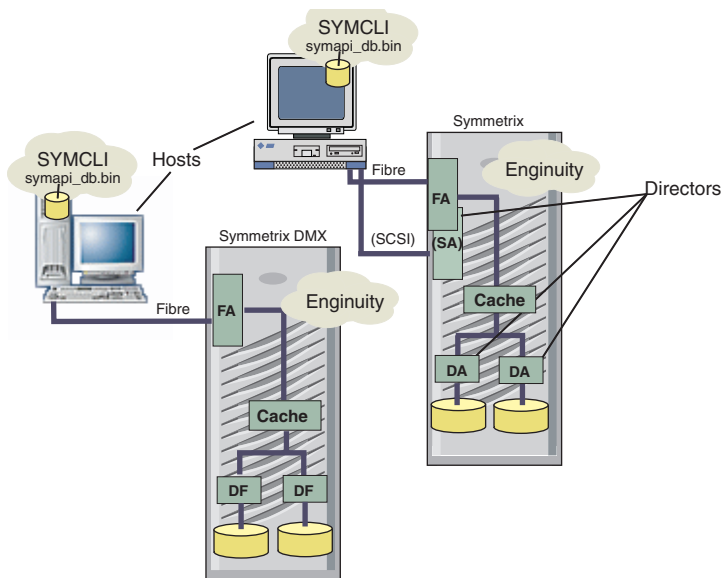


Figure 1-2 Symmetrix Components

Directors in the Enginuity Environment

In a Symmetrix array, a *front-end director* is 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.

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 Engenuity™ 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 Engenuity 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.

This chapter describes how to set up a SYMCLI environment for efficiency and introduces the basic operations and commands required to properly initiate a command line session. Details about avoidance and selection files and information on viewing and configuring Symmetrix logs. This chapter also provides some useful SYMCLI tips, and information on obtaining help.

- ◆ Command Line Environment Setup2-2
- ◆ Avoidance and Selection Files2-5
- ◆ Symmetrix Event Logs2-8
- ◆ SYMCLI Usage Tips2-16

Command Line Environment Setup

Prior to using the SYMCLI commands, you need to know some basic information on how to execute them. This chapter contains information on how to use the SYMCLI commands, how to reference devices, and how to apply command arguments and set the command line environment variables.

Note: Prior to using the SYMCLI commands, you need to run the `symcfg discover` command to build your configuration database. This needs to be done once after installation, and after any changes are made to your Symmetrix configuration. For more information on Discovery, refer to *Discovery* on page 3-1.

Set SYMCLI Path

Before you begin using SYMCLI, add the SYMCLI binary directory (`/usr/symcli/bin`) to your PATH environment variable.

For UNIX C shell, ensure the following SYMCLI directory is appended to variable PATH:

```
set path=(/usr/symcli/bin $path)
```

For UNIX Korn or Bourne shell, ensure the following SYMCLI directory is appended to variable PATH:

```
PATH=$PATH:/usr/symcli/bin
export PATH
```

On Windows, ensure the following SYMCLI path is current:

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

Starting and Stopping the Server

Before you begin to use SYMCLI on the host, you must start the SYMAPI service by using the following command executed from the server side:

```
symapisrv [-background]
           start
           [-service Servicename] | [-port Port]
```

For Windows, background mode does not apply.

For example, to start a SYMCLI session via a remote server with service name SYMAPI_SERVER, enter on your host:

```
symapisrv -serv SYMAPI_SERVER start
```

To stop the SYMCLI session via the remote server with service SYMAPI_SERVER, enter on your host or server:

```
symapisrv -serv SYMAPI_SERVER stop
```

Environment Variables

SYMCLI provides environment variables that can be preset to streamline and expedite your command line session. These environment variables can be preset to common argument values for a series of associated commands, which eliminates repeated key strokes for your given session. These are described in more detail in sections that follow.

To view a list of environment variables that can be set for a given SYMCLI session, enter:

```
symcli -env
```

To view the environment variables you currently have set, enter:

```
symcli -def
```

For information on configuring environment variables, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Options File for Optional Behavior Parameters

The options file, located in the SYMAPI configuration directory, contains behavior parameters that can be set to change defaults of certain behavior options within various SYMCLI components and associated SYMAPI calls. It can be used by advanced SYMCLI and SYMAPI users to customize and streamline command line coding to your specific environment.

For information about altering the optional behavior parameters, refer to the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Note: Altering parameters within this file are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use.

Examining SYMCLI Versions

To view the current SYMCLI software version running on your system, enter:

```
symcli
```

The following is an example output that identifies the SYMCLI version information. Your version information may vary from what is displayed here:

```
Symmetrix Command Line Interface (SYMCLI)
Version T6.0.0.22 (Edit Level: 595)
built with SYMAPI Version T6.0.0.22
(Edit Level: 595)
```

The Symmetrix Enginuity version can also be examined with:

```
symcfg list -v
```

Avoidance and Selection Files

The following are three *avoidance* files and one *selection* file that you can create and add to the SYMAPI configuration directory to limit the scope or change the performance of SYMCLI online commands; particularly the `symcfg discover` and `syming` commands:

- ◆ `symavoid`
- ◆ `inqfile`
- ◆ `gkavoid`
- ◆ `gkselect`

Note: These files and the following text are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use.

These files can be created to customize and streamline command line coding to your specific environment. If used, the files should be named as shown above and stored in the following SYMAPI configuration directory:

UNIX	<code>/var/symapi/config</code>
OpenVMS, Windows, AS/400, OS/390	<code>your_specific_installation_directory</code>



CAUTION

Be sure to delete these files when they are no longer needed because they can cause unexpected behavior and command limitations.

File Format

If you choose to use one or more of these files, they should be populated with physical device names (PdevNames) or Symmetrix IDs (SymmIDs), where each line contains a given entry—having only one device name or ID per line. SYMCLI ignores commented lines.

symavoid

The `symavoid` file affects the operation of `symcfg discover` so that it does not look for devices that belong to the Symmetrix arrays specified in this file. This may be useful if there are multiple Symmetrix arrays connected to the host that you want to disconnect from the SYMCLI. The Symmetrix avoidance file is formatted with 12-character Symmetrix IDs with one ID per line.



CAUTION

Be sure to delete these files when they are no longer needed because they can cause unexpected behavior and command limitations.

inqfile

The `inqfile` file configures calls to `symcfg discover` and `syminq`, to find only those PdevNames specified in this file. This may be useful if you want to limit the command(s) to view only certain Symmetrix devices from your host. The inquiry file is formatted with physical (host) device names with one PdevName (`/dev/rdsd/c2t0d2s2`) per line.



CAUTION

Be sure to delete these files when they are no longer needed because they can cause unexpected behavior and command limitations.

gkavoid

The `gkavoid` file affects calls to various online SYMCLI commands that use a gatekeeper to communicate to a Symmetrix array. A device whose PdevName matches any of the entries specified in the `gkavoid` file, will not be selected as a gatekeeper. This may be useful to ensure that certain devices are never used as gatekeepers. The gatekeeper avoidance file is formatted with physical device names, with one PdevName (for example, `/dev/rdsd/c2t0d1s1`) per line.

If a device is listed in both the `gkavoid` file and the `gkselect` file, the device will be avoided.

**CAUTION**

Be sure to delete these files when they are no longer needed because they can cause unexpected behavior and command limitations.

gkselect

To specify a set of devices to be the only devices used as gatekeepers, you can create the gatekeeper selection file, name it `gkselect`, and populate it with a list of selected physical device names.

The gatekeeper selection file, `gkselect`, affects all online SYMCLI command calls that utilize a gatekeeper to communicate with a Symmetrix array.

The gatekeeper selection file is formatted with physical device names, with one PdevName (for example, `/dev/rdisk/c2t0d1s1`) per line. Those devices whose PdevNames match any of the entries specified in the `gkselect` file will be the only devices used as gatekeepers. This may be useful to ensure specific devices are used as gatekeepers.

Creation of the `gkselect` file is optional, however, doing so will override any gatekeepers defined by the `symgate` command.

Note: If a device is listed in both the `gkavoid` file and the `gkselect` file, the device will be avoided.

**CAUTION**

Be sure to delete these files when they are no longer needed because they can cause unexpected behavior and command limitations.

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 2-1 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>
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>

Table 2-1 Log File Configuration Options (continued)

Logging Option	Configuration Description
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 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> CAUTION: 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 units, 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	2004 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				
Wed Dec 15 10:01:57	2004 Symm	Environment	Warning	04.0479.00
A Symmetrix power subsystem Environment sense cable is missing				

```

Wed Dec 15 11:02:29 2004 Symm Environment Warning 04.0479.00
A Symmetrix power subsystem Environment sense cable is missing

Wed Dec 15 12:02:29 2004 Symm Environment Warning 04.0479.00
A Symmetrix power subsystem Environment sense cable is missing

Wed Dec 15 13:03:42 2004 Symm Environment Warning 04.0479.00
A Symmetrix power subsystem Environment sense cable is missing

Wed Dec 15 14:04:13 2004 Symm Environment Warning 04.0479.00
A Symmetrix power subsystem Environment sense cable is missing

Wed Dec 15 16:04:13 2004 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:

```
syમેvent 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 2004:
  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 2004:
  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 has 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, 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 show -sid 442

                A U D I T   L O G   D A T A

Symmetrix ID           : 000185500442

Starting date          : 12/03/2004 13:15:06
Ending date            : 12/24/2004 16:52:06

Starting record number :      1
Ending record number   :     412
Total record count     :     412
```

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 list -sid 442 -rec 44 -n 5

      A U D I T   L O G   D A T A

Symmetrix ID           : 000185500442

Starting date          : 12/04/2004 19:48:37
Ending date            : 12/05/2004 10:35:24
. . .
. . .
```

Record Number	Date	Time	Application	Host	Function Class	Action Code
44	12/04/04	19:48:37	SYMCLI	api145	Base	RWEnable
45	12/04/04	19:54:28	SYMCLI	api145	Base	WRDisab
46	12/04/04	19:54:28	SYMCLI	api145	Base	WRDisab
47	12/05/04	10:35:18	SYMCLI	api145	CfgChg	Commit
48	12/05/04	10:35:24	SYMCLI	api145	CfgChg	Commit

The following is an example output of the `list` option for a specified Symmetrix array, beginning at record number 44 and displaying 1 record with verbose formatting:

```
# symaudit list -sid 442 -rec 44 -n 1 -v

      A U D I T   L O G   D A T A

Symmetrix ID           : 000185500442

Starting date          : 12/04/2004 19:48:37
Ending date            : 12/05/2004 10:35:24

Record Number          :          44
Records in Seq          :           1
Offset in Seq           :           1
Time                    : 12/04/04 19:48:37
Vendor ID               : EMC Corp
Application ID           : SYMCLI
Application Version      : 5.2.0.24
API Library              : SDK
API Version              : 5.2.0.24 (Edit Level: 406)
Host Name                : api145
OS Name                  : SunOS
OS Revision              : 5.6Generic
Client Host              :
Process ID               : 00023006
```

```

Task ID           : 00000000
Function Class    : Base
Action Code       : RWEnable
Text              : STARTING a Base Control
                  'RW_ENABLE' operation

```

- ◆ 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.

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 2-2 describes the various help options of the `symcli` command:

Table 2-2

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 2-3 lists and describes the various SYMCLI command line notations and conventions to identify objects.

Table 2-3 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

Table 2-3 Command Naming Conventions (continued)

Parameter/Ref.	Description	Examples
<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 Data3-2
- ◆ Synchronizing Configuration Updates with the Database3-4
- ◆ SYMAPI Configuration Database File3-5

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.

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, or just CLARiiON arrays by specifying the `-clariion` option. For more information about the Symmetrix configuration database, refer to *SYMAPI Configuration Database File* on page 3-5.

`symcfg discover` scans all SCSI buses on the host, not just those connected to Symmetrix and CLARiiON 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).

Scanning the Hardware

When a set of devices accessible by a given host has changes, a `symcfg scan` command should be issued to activate 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.

Synchronizing Configuration Updates with the Database

To update configuration data about previously discovered arrays, while maximizing performance, you should synchronously update those configuration changes with the configuration database. 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 if a configuration change has occurred, the configuration 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 3-1 Database Synchronization Options

Array	Command and Options	Description
Symmetrix	<pre>symcfg sync [-sid SymmID] [-rdf -bcv -local -dirsts -snap -cfgmgr]</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
CLARiiON	<pre>symcfg -clariion 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 Symmetrix configuration database, `symapi_db.bin`, described next. Throughout the remainder of this manual it will be referred to as the Symmetrix configuration database.

Note: For information about discovering your storage environment and building your Symmetrix configuration database, refer to *Discovering Configuration and Status Data* on page 3-2.

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 currently in use, 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 `symbbackup_db.bin`, enter:

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

To perform the same operation on Windows, enter:

```
set SYMCLI_DB_FILE=C:\Program  
Files\EMC\Symapi\db\symapi_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 : 6.0.X.X (Edit Level: 595)
Version of SYMAPI Library which wrote the DB      : 6.0.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 2004
Time Any Device Group Was Last Modified  : Tue Nov 16 15:17:09 2004

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.

Configuration Data

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

◆ Levels of Configuration Data	4-2
◆ SCSI Inquiries	4-3
◆ Array Data.....	4-7
◆ Symmetrix Devices	4-20
◆ CLARiiON Devices	4-36
◆ Disk-Level Data	4-37

Levels of Configuration Data

Solutions Enabler SYMCLI allows you to gather and display various levels of information about your storage environment. SYMCLI allows you to focus your queries to 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, 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 further details about the devices specifically attached (locally or remotely) to a given array (or all arrays). Device-level data includes capacity, cache, emulation, configuration, group, and usage information that is critical to all other storage management operations.

Disk-Level

In addition, you can return 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 issue SCSI INQUIRY, and optionally SCSI READ CAPACITY, on host HBAs and one or all locally attached physical devices.

Scope of Inquiries

You can limit the scope of an inquiry to Symmetrix, CLARiiON, StorageWorks, or HDS devices. When specifying a specific array, options are provided to control the data returned and its presentation, as specified 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.

SCSI Inquiry Device Information: Examples

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	Num	Vendor	ID	Rev	Ser Num	Cap (KB)
/dev/rdisk/c0t1d0s2		N/A	SEAGATE	ST318203LSUN18G	034A	99414938	17689266
/dev/rdisk/c2t0d0s2		0017	EMC	SYMMETRIX	5670	7100017630	1056000

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/rdisk/c2t0d0s2		000187900771	5670	7100017630	1056000
/dev/rdisk/c2t0d1s2		000187900771	5670	7100018630	1056000

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/rdisk/c2t0d0s2		000187900771	5670	7100017630	
/dev/rdisk/c2t0d1s2		000187900771	5670	7100018630	

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/rdisk/c2t0d0s2	0017	000187900771	6006048000018790077153594D303137
/dev/rdisk/c2t0d1s2	0018	000187900771	6006048000018790077153594D303138
/dev/rdisk/c2t0d2s2	0019	000187900771	6006048000018790077153594D303139
/dev/rdisk/c2t0d3s2	001A	000187900771	6006048000018790077153594D303141

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 spaces with colons:

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

Device		Device	
Name	Num	Array ID	WWN
/dev/rdisk/emcpower393c 3:59:4D:32:43:41	02CA	000187940256	60:06:04:80:00:01:87:94:02:56:5
/dev/rdisk/emcpower394c 3:59:4D:32:43:38	02C8	000187940256	60:06:04:80:00:01:87:94:02:56:5
/dev/rdisk/emcpower395c 3:59:4D:32:43:39	02C9	000187940256	60:06:04:80:00:01:87:94:02:56:5
/dev/rdisk/emcpower396c 3:59:4D:32:43:43	02CC	000187940256	60:06:04:80:00:01:87:94:02:56:5

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. If you want, this request can be limited to list only the Fibre (`-fibre`), SCSI (`-scsi`), or iSCSI (`-iscsi`) HBAs. In addition, the `-colon` option can be used to return a the device WWNs spaced with colons.

In addition, the `-snia` option allows you to return HBA information via the native SNIA API rather than by issuing a SCSI inquiry. The returned data will be exactly the same as the data returned via a 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.

Array Data

When you begin to query your storage environment configuration data, you must first identify the arrays in your storage environment. Each array has a unique serial number. On Symmetrix arrays, they are called a Symmetrix ID (SID) and on CLARiiON arrays it is known as a CLARiiON ID (CID).

List of Symmetrix Arrays

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

For Symmetrix arrays, you can drill deeper into the array using SYMCLI 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 CLARiiON IDs of arrays connected to your host, with one entry for each CLARiiON array accessible from the host, enter:

```
symcfg list -clariion
```

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

```
symcfg list
```

An example output appears below:

```

S Y M M E T R I X

SymmID      Attachment  Model      Mcode      Cache      Num Phys      Num Symm
              Version      Size (MB)   Devices    Devices
000187900771 Local        800-M2     5670       32768      57           3316
000000005092 Remote      8530       5568        6144       0            525
000000005217 Remote      8230       5568        2048       0            498
```

Symmetrix Configuration Data

To obtain detailed configuration and status data for all the Symmetrix arrays enter:

```
symcfg list -v
```

The following example output displays the details for one Symmetrix array that would be included in the verbose list of all arrays:

```
Symmetrix ID: 000187900771 (Local)
Time Zone   : EST
```

```

Product Model           : 800-M2
Symmetrix ID            : 000187900771

Microcode Version (Number) : 5670 (16260000)
Microcode Date           : 09.16.2004

Microcode Patch Date     : 09.16.2004
Microcode Patch Level    : 78

Cache Size               : 32768 (MB)
# of Available Cache Slots : 532980
Max # of System Write Pending Slots : 130000
Max # of DA Write Pending Slots : 65000
Max # of Device Write Pending Slots : 600

Symmetrix Total Operating Time : 114 days, 05:33:41
Symmetrix Power ON Time       : Fri Aug 13 13:44:36 2004
Symmetrix Last IPL Time (Cold) : Wed Oct 27 13:41:39 2004
Symmetrix Last Fast IPL Time (Hot) : Fri Dec 3 11:59:53 2004

Host DB Sync Time          : Fri Dec 3 08:56:04 2004
Symmetrix CLI (SYMCLI) Version : T6.0.0.19 (Edit Level: 595)
Built with SYMAPI Version    : T6.0.0.19 (Edit Level: 595)
SYMAPI Run Time Version     : T6.0.0.19 (Edit Level: 595)

Number of Configured (Sym) Devices : 3316
Number of Visible (Host) Devices : 57
Number of Configured Actual Disks : 41
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      : A7D1B2
Switched RDF Configuration State      : Enabled
Concurrent RDF Configuration State    : Enabled
Dynamic RDF Configuration State       : Enabled
Concurrent Dynamic RDF Configuration : Disabled
RDF Data Mobility Configuration State : Disabled
Access Control Configuration State    : Disabled
Device Masking (VCM) Config State     : Disabled
VCMdb Access Restricted State         : Disabled
Multi LRU Device Assignment           : BY_NUMBER
Disk Group Assignments                : In Use

Parity Raid Configuration             : RAID-S (3+1)
Raid-5 Configuration                 : RAID-5 (3+1)
PAV Mode                             : NoPAV
PAV Alias Limit                      : 0

SRDF/A Maximum Host Throttle (Secs)  : 0
SRDF/A Maximum Cache Usage (Percent) : 100

```

< . . . >

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 000000006196, enter:

```
symcfg list -applications -sid 6196
```

An example output appears below:

```
Symmetrix ID      : 000000006196
```

Host		Application			
Node Name	IP Address	ID	Vendor ID	Version	Attr
hst145	162.23.60.135	SYMCLI	EMC Corp	5.2.0.24	-
		SYMCLI	EMC Corp	5.2.0.23	-
		STORCLI	EMC Corp	5.2.0.18	-
		SYMCLI	EMC Corp	5.2.0.20	-
		SYMCLI	EMC Corp	5.2.0.16	-

< . . . >

Legend for Attribute(s):

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

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

Viewing Hosts with Symmetrix Access

To view all the hosts that have accessed the Symmetrix arrays to which your host is connected, enter:

```
symcfg list -connections
```

An example output appears below:

```
Symmetrix ID : 000000006196
```

Symmetrix		Host				
Director	Port	Node Name	IP Address	HW Type	OS Name	OS Revision
DF-1A	N/A	hst201	162.23.60.199	INTEL	WinNT-SP	4.0.1381
FA-2A	1	hst145	162.23.60.135	sun4u	SunOS	5.6

```
Symmetrix ID : 000184600063
```

Symmetrix		Host				
Director	Port	Node Name	IP Address	HW Type	OS Name	OS Revision
FA-14A	0	hst145	172.23.60.135	sun4u	SunOS	5.6

```
<. . .>
```

In the example above, the display of the host connections are sorted according to each Symmetrix. 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
```

An example output appears below:

Host				Symmetrix		
Node Name	IP Address	OS Name	OS Revision	ID	Director	Port
hst145	162.23.60.135	SunOS	5.6	000000006196	FA-2A	1
				000184600063	FA-14A	0
hst147	172.23.63.157	LINUX	2.4.18-10s	000184600282	FA-14A	0
hst148	172.23.60.188	LINUX	2.4.19-64G	000184600282	FA-14A	0
hk184600282	172.20.125.150	WinNT-SP	4.0.1381	000184600282	DA-1A	0
hst201	162.23.60.199	WinNT-SP	4.0.1381	000000006196	DF-1A	0

In addition, you can focus the return data to a specific Symmetrix array, by specifying a specific Symmetrix ID (-sid) in either of these command line entries.

Viewing Storage Capacities to Hosts

The -connections option can be used in conjunction with the -capacity option to view the storage capacities of all the hosts that have accessed one or all Symmetrix arrays to which your host is connected.

For example, to list the allocated storage capacities of all the host names that have worked with Symmetrix 6196, enter:

```
symcfg list -connections -capacity -sid 6196
```

An example output appears below:

Host		Symmetrix		Capacity (GB)		
Node Name	ID	Director	Port	Mapped STDs	Mapped BCVs	Paired BCVs
hst201	000000006196	DF-1A	0	0.0	0.0	0.0
hst201 totals:				0.0	0.0	0.0
hst145	000000006196	FA-2A	1	0.7	0.0	0.0
hst145	000000006196	FA-2B	1	7.0	0.0	0.0
hst145 totals:				7.7	0.0	0.0

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

Viewing Network Services

To view the available network services in the client/SYMAPI server connection, enter:

```
symcfg list -services
```

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 indicate 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.

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
```

The following is an example output for one Symmetrix array:

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

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
FA-2A	02A	2	2	FibreChannel	Online
EA-15A	15A	15	15	ESCON	Online
DF-16A	16A	16	16	DISK	Online
DF-1B	01B	17	1	DISK	Online
FA-2B	02B	18	2	FibreChannel	Online
EA-15B	15B	31	15	ESCON	Online
DF-16B	16B	32	16	DISK	Online
DF-1C	01C	33	1	DISK	Online
RF-2C	02C	34	2	RDF-BI-DIR	Online
EA-15C	15C	47	15	ESCON	Online
DF-16C	16C	48	16	DISK	Online
DF-1D	01D	49	1	DISK	Online
RF-2D	02D	50	2	RDF-BI-DIR	Online
EA-15D	15D	63	15	ESCON	Online
DF-16D	16D	64	16	DISK	Online

```
< . . . >
```

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:

```
symcfg -sa all list
```

The following is an example output for one Symmetrix array:

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

```

      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-2A   02A       2        2    FibreChannel   Online
FA-2B   02B      18        2    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
```

The following is an example output for one Symmetrix array:

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

```

      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  Port Status      Connection Status
      Type           Status    P0 P1 P2 P3      P0 P1 P2 P3
FA-2A  FibreChannel   Online    ON  ON  N/A N/A    -  X  -  -
FA-2B  FibreChannel   Online    ON  ON  N/A N/A    -  -  -  -
< . . . >
```

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 000000006196, enter:

```
symcfg list -dir all -address -sid 6196
```

The following is an example output:

Symmetrix ID: 000000006196

Director			Device Name		Attr	Address		
Ident	Symbolic	Port	Sym	Physical		VBUS	TID	LUN
FA-2A	02A	0	0000	Not Visible	VCM	0	0	000
FA-2A	02A	1	0000	/dev/rdisk/clt0d0s2	VCM	0	0	000
			0040	/dev/rdisk/clt0d1s2		0	0	001
			0041	/dev/rdisk/clt0d2s2		0	0	002
			0042	/dev/rdisk/clt0d3s2		0	0	003

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 6196
```

The following is an example output for one Symmetrix array:

Symmetrix ID: 000000006196

Director			Device Name		Attr	Address		
Ident	Symbolic	Port	Sym	Physical		VBUS	TID	LUN
FA-2A	02A	0	0000	Not Visible	VCM	0	0	000
			-	AVAILABLE		0	0	001 *
FA-2A	02A	1	0000	/dev/rdisk/clt0d0s2	VCM	0	0	000
			0040	/dev/rdisk/clt0d1s2		0	0	001
			0041	/dev/rdisk/clt0d2s2		0	0	002
			-	AVAILABLE		0	0	005 *

< . . . >

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.

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, or only GNS locks.

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 to 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
```

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	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 to all possible locks use the keyword `ALL`, for RDF-related locks use the keyword `RDF`, for RDF-A-related locks use the keyword `RDFA`, and for GNS related locks use the keyword `GNS` with the `-lockn` option. For example, to return a list of all host-visible Symmetrix arrays (local and remote) along with details about any RDF exclusive locks that are currently locked, enter:

```
symcfg list -lockn RDF
```

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 098712341357, enter:

```
symcfg -sid 098712341357 -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.

For information about device external locks that only target specific devices, refer to *Releasing Device External Locks* on page 4-35.

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 043033 offline, enter:

```
symcfg offline -RA 12 -sid 043033
```

To bring the director back online, enter:

```
symcfg online -RA 12 -sid 043033
```

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 043033 offline, enter:

```
symcfg offline -SA 12 -P 1 -sid 043033
```



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).

In this case, 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 043033 back online, enter:

```
symcfg online -SA 12 -P 1 -sid 043033
```

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 *hyper-volume* 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 hyper-volumes, 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 (blocks ÷ 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.)

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 4-1 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 5, <i>Gatekeeper Devices</i> .
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 Enginuity 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> .

Table 4-1 Device Types (continued)

Device Type	Description
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> .
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 Device Masking 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 Optimization 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/rdisk/clt0d0s2	0000	02A:1	01C:C6	Unprotected	N/Grp'd	VCM WD	47
/dev/rdisk/clt0d1s2	0040	02A:1	16A:D4	Unprotected	N/Grp'd	RW	188
/dev/rdisk/clt0d2s2	0041	02A:1	01D:D4	Unprotected	N/Grp'd	RW	188
/dev/rdisk/clt0d3s2	0042	02A:1	16B:C4	Unprotected	N/Grp'd	RW	188
/dev/rdisk/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).

For example, for a list of Symmetrix device names on Symmetrix 000000006196, enter:

```
symdev list -sid 0771
```

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.

The following example output is truncated:

Symmetrix ID: 000000000771

Device Name		Directors		Device			
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)	
0000	/dev/rdisk/clt0d0s2	02A:1 01C:C6	Unprotected	N/Grp'd	VCM WD	47	
0001	Not Visible	????:? 16B:C6	RDF1	N/Grp'd	RW	469	
0002	Not Visible	????:? 01A:C6	RDF1	N/Grp'd	RW	469	
0003	Not Visible	????:? 16C:D6	RDF1	N/Grp'd	RW	469	
0004	Not Visible	????:? 01B:D6	RDF1	N/Grp'd	RW	469	
0005	Not Visible	????:? 16D:C6	RDF2	N/Grp'd	RW	469	
0006	Not Visible	????:? 01C:D3	RDF2	N/Grp'd	RW	469	
0007	Not Visible	????:? 16A:C3	RDF2	N/Grp'd	RW	469	
<. . .>							

Example Output Verbose Option

The verbose option (-v) with either `symdev list` or `sympd list` will return details about each device.

The following is an example of the detailed output returned for a host-visible device using the `[symdev | sympd] list -v` command entry:

```
Device Physical Name      : /dev/rdisk/clt0d102s2

Device Symmetrix Name     : 0D8C
Device Serial ID          : 71D8C330
Symmetrix ID              : 000187900771

Attached BCV Device       : N/A

Attached VDEV TGT Device  : N/A

Vendor ID                 : EMC
Product ID                : SYMMETRIX
Product Revision          : 5670
Device WWN                : 6006048000018790077153594D443843
Device Emulation Type     : FBA
Device Defined Label Type : N/A
Device Defined Label      : N/A
Device Sub System Id      : 0x000E

Device Block Size         : 512

Device Capacity
{
```

```

Cylinders      :      5000
Tracks        :      75000
512-byte Blocks :    4800000
MegaBytes     :        2344
KiloBytes     :    2400000
}

```

```

Device Configuration : Unprotected      (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 : None

```

```

Device Service State : Normal

```

```

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

```

```

Front Director Paths (2):

```

```

{
-----
                                POWERPATH  DIRECTOR  PORT              LUN
                                -----
PdevName                      Type        Type Num    Sts  VBUS  TID  SYMM  Host
-----
/dev/rdisk/clt0d102s2        N/A      FA   01C:0  RW   000  00  066  066
Not Visible                  N/A      FA   16D:1  RW   000  00  061  N/A
}

```

```

Meta Configuration : Concatenated

```

```

Meta Device Members (100):

```

```

{
-----
                                BCV  DATA              RDF  DATA
                                -----
Sym   Cap   Std Inv BCV Inv Pair              R1 Inv R2 Inv Pair
Dev   (MB)  Tracks Tracks State              Tracks Tracks State
-----
--> 0D8C   23      -   -   N/A              -   -   N/A
    0D8D   23      -   -   N/A              -   -   N/A
    0D8E   23      -   -   N/A              -   -   N/A
    0D8F   23      -   -   N/A              -   -   N/A
    0D90   23      -   -   N/A              -   -   N/A
    0D91   23      -   -   N/A              -   -   N/A
    0D92   23      -   -   N/A              -   -   N/A
    0D93   23      -   -   N/A              -   -   N/A
}

```

Configuration Data

0D94	23	-	-	N/A	-	-	N/A
0D95	23	-	-	N/A	-	-	N/A
0D96	23	-	-	N/A	-	-	N/A
0D97	23	-	-	N/A	-	-	N/A
0D98	23	-	-	N/A	-	-	N/A
0D99	23	-	-	N/A	-	-	N/A
0D9A	23	-	-	N/A	-	-	N/A
0D9B	23	-	-	N/A	-	-	N/A
0D9C	23	-	-	N/A	-	-	N/A
0D9D	23	-	-	N/A	-	-	N/A
0D9E	23	-	-	N/A	-	-	N/A
0D9F	23	-	-	N/A	-	-	N/A
0DA0	23	-	-	N/A	-	-	N/A
0DA1	23	-	-	N/A	-	-	N/A
0DA2	23	-	-	N/A	-	-	N/A
0DA3	23	-	-	N/A	-	-	N/A
0DA4	23	-	-	N/A	-	-	N/A
0DA5	23	-	-	N/A	-	-	N/A
0DA6	23	-	-	N/A	-	-	N/A
0DA7	23	-	-	N/A	-	-	N/A
0DA8	23	-	-	N/A	-	-	N/A
0DA9	23	-	-	N/A	-	-	N/A
0DAA	23	-	-	N/A	-	-	N/A
0DAB	23	-	-	N/A	-	-	N/A
0DAC	23	-	-	N/A	-	-	N/A
0DAD	23	-	-	N/A	-	-	N/A
0DAE	23	-	-	N/A	-	-	N/A
0DAF	23	-	-	N/A	-	-	N/A
0DB0	23	-	-	N/A	-	-	N/A
0DB1	23	-	-	N/A	-	-	N/A
0DB2	23	-	-	N/A	-	-	N/A
0DB3	23	-	-	N/A	-	-	N/A
0DB4	23	-	-	N/A	-	-	N/A
0DB5	23	-	-	N/A	-	-	N/A
0DB6	23	-	-	N/A	-	-	N/A
0DB7	23	-	-	N/A	-	-	N/A
0DB8	23	-	-	N/A	-	-	N/A
0DB9	23	-	-	N/A	-	-	N/A
0DBA	23	-	-	N/A	-	-	N/A
0DBB	23	-	-	N/A	-	-	N/A
0DBC	23	-	-	N/A	-	-	N/A
0DBD	23	-	-	N/A	-	-	N/A
0DBE	23	-	-	N/A	-	-	N/A
0DBF	23	-	-	N/A	-	-	N/A
0DC0	23	-	-	N/A	-	-	N/A
0DC1	23	-	-	N/A	-	-	N/A
0DC2	23	-	-	N/A	-	-	N/A
0DC3	23	-	-	N/A	-	-	N/A
0DC4	23	-	-	N/A	-	-	N/A
0DC5	23	-	-	N/A	-	-	N/A
0DC6	23	-	-	N/A	-	-	N/A
0DC7	23	-	-	N/A	-	-	N/A

0DC8	23	-	-	N/A	-	-	N/A
0DC9	23	-	-	N/A	-	-	N/A
0DCA	23	-	-	N/A	-	-	N/A
0DCB	23	-	-	N/A	-	-	N/A
0DCC	23	-	-	N/A	-	-	N/A
0DCD	23	-	-	N/A	-	-	N/A
0DCE	23	-	-	N/A	-	-	N/A
0DCF	23	-	-	N/A	-	-	N/A
0DD0	23	-	-	N/A	-	-	N/A
0DD1	23	-	-	N/A	-	-	N/A
0DD2	23	-	-	N/A	-	-	N/A
0DD3	23	-	-	N/A	-	-	N/A
0DD4	23	-	-	N/A	-	-	N/A
0DD5	23	-	-	N/A	-	-	N/A
0DD6	23	-	-	N/A	-	-	N/A
0DD7	23	-	-	N/A	-	-	N/A
0DD8	23	-	-	N/A	-	-	N/A
0DD9	23	-	-	N/A	-	-	N/A
0DDA	23	-	-	N/A	-	-	N/A
0ddb	23	-	-	N/A	-	-	N/A
0DDC	23	-	-	N/A	-	-	N/A
0DDD	23	-	-	N/A	-	-	N/A
0DDE	23	-	-	N/A	-	-	N/A
0DDF	23	-	-	N/A	-	-	N/A
0DE0	23	-	-	N/A	-	-	N/A
0DE1	23	-	-	N/A	-	-	N/A
0DE2	23	-	-	N/A	-	-	N/A
0DE3	23	-	-	N/A	-	-	N/A
0DE4	23	-	-	N/A	-	-	N/A
0DE5	23	-	-	N/A	-	-	N/A
0DE6	23	-	-	N/A	-	-	N/A
0DE7	23	-	-	N/A	-	-	N/A
0DE8	23	-	-	N/A	-	-	N/A
0DE9	23	-	-	N/A	-	-	N/A
0DEA	23	-	-	N/A	-	-	N/A
0DEB	23	-	-	N/A	-	-	N/A
0DEC	23	-	-	N/A	-	-	N/A
0DED	23	-	-	N/A	-	-	N/A
0DEE	23	-	-	N/A	-	-	N/A
0DEF	23	-	-	N/A	-	-	N/A

```

-----
      2344      -      -      -      -
}

```

Mirror Set Type : [Data,N/A,N/A,N/A]

Mirror Set DA Status : [RW,N/A,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] : [16B, C, 6]
Disk Director Volume Number : 685 (0x2AC)
Hyper Number              : 169
Disk Capacity             : 139814m
Disk Group Number        : 0
}

Device Physical Name      : /dev/rdisk/clt0d103s2

Device Symmetrix Name     : 0DF0
Device Serial ID         : 71DF0330
Symmetrix ID             : 000187900771

Device Group Name        : steve
Device Logical Name      : DEV028

Attached BCV Device      : N/A

Attached VDEV TGT Device : N/A

Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5670
Device WWN               : 6006048000018790077153594D444630
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x000E

Device Block Size        : 512

Device Capacity
{
  Cylinders      : 660
  Tracks         : 9900
  512-byte Blocks : 633600
  MegaBytes      : 309
  KiloBytes      : 316800
}

Device Configuration     : Unprotected (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 : None

Device Service State : Normal

Device Status : Ready (RW)

Device SA Status : Ready (RW)

Front Director Paths (1):

	POWERPATH	DIRECTOR	PORT	LUN				
PdevName	Type	Type	Num	Sts	VBUS	TID	SYMM	Host

/dev/rdisk/c1t0d103s2	N/A	FA	01C:0	RW	000	00	067	067
}								

Meta Configuration : Concatenated

Meta Device Members (11) :

BCV DATA						RDF DATA				
Sym	Cap	Std	Inv	BCV	Inv	Pair	R1	Inv	R2	Inv
Dev	(MB)	Tracks	Tracks			State	Tracks	Tracks		State

--> 0DF0	28	-	-	-	-	N/A	-	-	-	N/A
0DF1	28	-	-	-	-	N/A	-	-	-	N/A
0DF2	28	-	-	-	-	N/A	-	-	-	N/A
0DF3	28	-	-	-	-	N/A	-	-	-	N/A
0DF4	28	-	-	-	-	N/A	-	-	-	N/A
0DF5	28	-	-	-	-	N/A	-	-	-	N/A
0DF6	28	-	-	-	-	N/A	-	-	-	N/A
0DF7	28	-	-	-	-	N/A	-	-	-	N/A
0DF8	28	-	-	-	-	N/A	-	-	-	N/A
0DF9	28	-	-	-	-	N/A	-	-	-	N/A
0DFA	28	-	-	-	-	N/A	-	-	-	N/A

	309	-	-	-	-		-	-	-	
}										

Mirror Set Type : [Data,N/A,N/A,N/A]

Mirror Set DA Status : [RW,N/A,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]      : [16B, C, 8]
Disk Director Volume Number         : 859 (0x35A)
Hyper Number                         : 171
Disk Capacity                       : 139814m
Disk Group Number                   : 0
}

```

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

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
-----	----	----	----
Unprotected	1140	1	N/A
2-Way Mir	397	N/A	N/A
3-Way Mir	1	N/A	N/A
RAID-S	3	N/A	N/A
RAID-5	1115	1	N/A
RDF2	26	N/A	N/A
RDF2+R-S	3	N/A	N/A
BCV	595	N/A	N/A
DRV	2	N/A	N/A
VDEV	8	N/A	N/A
COVD	26	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 `sympd`

When calling `sympd 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.

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`).

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` 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. When calling this command, a number of filters 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 is the syntax for `symdev list`:

```
symdev list [-sid SymmID] [-offline] [-v] [-resv]
            [-wwn] [-SA <#|ALL>] [-P #] [-CAP #]
            [-N #] [-cyl] [-scsi] [-fibre]
            [-held] [-RADIANT]
            [-noport|-multiport|-firstport]
            [-RANGE <SymDevStart:SymDevEnd>]
            [-R1] [-R2] [-bcv|-nobcv|-drv] [-vcm]
            [-meta] [-hotspare] [-dynamic] [-worm]
            [-vdev] [-savedev] [-raids] [-raid5]
            [-rdfa] [-disk_group nn] [pd]
```

In all `symdev list` 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.

Filter by Director

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`).

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. Where as, the hyper capacity (Hyper Num Cap) displayed for each member is the physical capacity, which included 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                      : 600604800000000619653594D483842
```

```
. . .
```

```
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 :IT	DA Vol#	Hyper Num	Cap(MB)	Member Num	Status	Spare Status	Disk 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 Hyper Volume

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 4-24.

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, TimeFiner, 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	6006016095E30D0066D61F3BAF10D911

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 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 `-hotspares` 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 (01A):

```
symdisk list -sid 000183600335 -da 01A
```

```
Symmetrix ID      : 000183600335
Disks Selected    : 8
```

Ident	Symb	Int	TID	Vendor	Type	Hypers	Capacity(MB)	
							Total	Free
DA-1A	01A	C	0	SEAGATE	SP_D_1_	3	17366	14273
DA-1A	01A	C	1	SEAGATE	SP_D_1_	3	17366	14273
DA-1A	01A	C	2	SEAGATE	SP_D_1_	7	17366	12205
DA-1A	01A	C	3	SEAGATE	SP_D_1_	8	17366	9116
DA-1A	01A	D	0	SEAGATE	SP_D_1_	3	17366	14273
DA-1A	01A	D	1	SEAGATE	SP_D_1_	8	17366	11173
DA-1A	01A	D	2	SEAGATE	SP_D_1_	7	17366	10148
DA-1A	01A	D	3	SEAGATE	SP_D_1_	8	17366	11173

The following example shows details about the specified disk (01A:C0):

```
symdisk show 01A:C0 -sid 000183600335
```

```
Symmetrix ID      : 000183600335
Director         : DA-1A
Interface        : C
Target ID        : 0
```

```
Vendor ID         : SEAGATE
Product ID        : SX118202LC
Product Revision  : SP_D_1_
Serial ID         : LK383257
```

```
Disk Blocks       : 35566479
Block Size        : 512
Disk Capacity (MB) : 17366
```

```
Hypers (3):
```

```
{
#  Vol Emulation      Dev  Type      Mirror Status      Cap(MB)
--  --  -
1   1  FBA            0000 Data        1    Ready        1031
2   2  FBA            0078 Data        1    Ready        1031
3   3  FBA            008C Data        1    Ready        1031
}
```

Note that 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 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.

Gatekeeper Devices

This chapter describes the function of gatekeepers and how to manage them in your system.

- ◆ Overview5-2
- ◆ SYMCLI Gatekeeper Management.....5-6

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*.

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 5-2.

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.

Undefining 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 5-8).

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
```

After you disassociate the gatekeeper from the device group, SYMCLI chooses the first available gatekeeper device in the gatekeeper list before it routes a command to the Symmetrix via a gatekeeper device.

Grouping Devices

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

- ◆ Overview6-2
- ◆ Device Groups6-4
- ◆ Composite Groups6-20

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.

Groupings

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.
- ◆ 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 6-20.
 - 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 6-21 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*.

Device Groups

A device group is a user-defined object comprised of devices that belong to a single Symmetrix array and a single RA group. 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.

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.

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.

SRDF Devices

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 RDF, 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.

BCV Devices

You can associate Business Continuance 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.

Virtual Devices

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,

-
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.

TimeFinder/Snap control operations can be performed on any virtual device in a device group.

Gatekeeper Devices

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.

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.

For more information about associating a gatekeeper device with a group, refer to Chapter 5, *Gatekeeper Devices*.

How to Create a Device Group

A device group is created using the `symdg` command. When you create a device group, you assign it a name. Additionally, when the group is part of an RDF configuration, you must specify whether or not its members are operating as source devices (RDF1 type) to the remote Symmetrix or operating as remote target devices (RDF2 type).

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

```
symdg -type RDF1 create prod
```

Adding Devices

SymCLI provides various commands to add standard devices (STD) or virtual devices (VDEV) 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 (`symldg`), or a list of devices (`symldg`) to that group. For more information on adding a list of devices, refer to *Exporting and Importing Device Lists* on page 6-14.

Adding a Single Device

Use `symld` 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 (`pd`) or the Symmetrix device name (`dev`).

For example, 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.

For example, to add a single device, `/dev/rhdisk32` to a device group named `prod` and assign the logical device name `temp1`, enter:

```
symld -g prod add pd /dev/rhdisk32 temp1
```

SRDF 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, as specified by the device group type
- ◆ Must have the same Symmetrix RDF group number

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
```

Adding All Ungrouped Devices

Use `symld` to add all un-grouped devices on a Symmetrix array to a device group. By default, this command will add all standard 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:

- ◆ 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 (`-vdev`)
- ◆ All ungrouped devices belonging to a specific RDF group number (`-RDFG #`)

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

```
symld -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 *Environment Variables* on page 2-3.

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	Number of		
					GKs	BCVs	VDEVs
rdf2	RDF2	N/A	000187900041	40	0	40	3
vsn	REGULAR	N/A	000187900041	99	0	0	99
hop32dev	RDF1	N/A	000187900041	6	0	6	0
vsnap3	REGULAR	N/A	000187900041	1	0	1	1

This command returns a list of all device group names, group type, Symmetrix ID, and the number of standard, BCV, gatekeeper, and virtual devices.

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 `hop32dev`, enter:

```
symdg show hop32dev
```

The following details may be displayed:

- ◆ The type of the device group (REGULAR, RDF1, or RDF2).
- ◆ Whether the device group in GNS.
- ◆ The ID of the Symmetrix to which the device group belongs.
- ◆ Group creation time.
- ◆ The number of standard devices in the device group.
- ◆ The number of gatekeeper devices associated with the device group.
- ◆ The number of Business Continuance Volume (BCV) devices associated with the device group.
- ◆ Number of remote BCV devices.
- ◆ List of all standard devices in the device group.
- ◆ List of all grouped gatekeeper devices associated with a device group.
- ◆ List of all local/remote BCV devices associated with the device group.
- ◆ Group-wide RDF information, if this is an RDF device group.
- ◆ Virtual device information.
- ◆ SRDF/A information.

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

Group Name: hop32dev

```

Group Type                : RDF1
Device Group in GNS       : Yes
Valid                     : Yes
Symmetrix ID              : 000187900041
Group Creation Time       : Wed Dec 3 15:07:49 2003
Vendor ID                 : EMC Corp
Application ID            : SYMCLI

```

```

Number of STD Devices in Group      : 6
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) : 6
Number of Remotely-associated BCV's (BCV RDF) : 0
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 6

```

Standard (STD) Devices (6):

```

{
-----
LdevName          PdevName          Sym      Att.  Sts      Cap
Dev              (MB)
-----
DEV001            N/A              0002      RW      1031
DEV002            N/A              0003      RW      1031
DEV003            N/A              0004      RW      1031
DEV004            N/A              0005      RW      1031
DEV005            N/A              0006      RW      1031
DEV006            N/A              0007      RW      1031
}

```

BCV Devices Remotely-associated (STD RDF) (6):

```

{
-----
LdevName          PdevName          Sym      Att.  Sts      Cap
Dev              (MB)
-----
RBCV001           N/A              0048      NR      1031
RBCV002           N/A              0049      NR      1031
RBCV003           N/A              004A      NR      1031
RBCV004           N/A              004B      NR      1031
RBCV005           N/A              004C      NR      1031
RBCV006           N/A              004D      NR      1031
}

```

RBCV Devices Remotely-associated (RBCV RDF) (6):

```
{
-----
```

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)
RRBCV001	N/A	0030		RW	1031
RRBCV002	N/A	0031		RW	1031
RRBCV003	N/A	0032		RW	1031
RRBCV004	N/A	0033		RW	1031
RRBCV005	N/A	0034		RW	1031
RRBCV006	N/A	0035		RW	1031

```
}
```

Device Group RDF Information

```
{
RDF Type : R1
RDF (RA) Group Number : 1 (A)

Remote Symmetrix ID : 000187900042

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 : Disabled

Device RDF State : Ready (RW)
Remote Device RDF State : Ready (RW)

RDF Pair State ( R1 <=> R2 ) : Split

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

Device Group RBCV RDF Information

```

{
  RDF Type                               : R1
  RDF (RA) Group Number                   : 2                               (B)

  Remote Symmetrix ID                     : 000187900043

  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                        : Not Ready                       (NR)
  Remote Device RDF State                 : Write Disabled                (WD)

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

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

```

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.

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

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

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 6-15) 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 6-19.

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 6-14).

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.

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 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
```

Adding Devices from a Device Group to a Composite Group

Use the following syntax to add devices from an existing device group to an existing composite group:

```
symdg [-bcv|-vdev|-rbcv|-rrbcv|-brbcv|-nobcv]
      [-apidb | -ppath | -rdf_consistency] [-force]
      [-rename] dg2cg DgName CgName
```

The following additional options are available:

- ◆ `-vdev` adds only the VDEVs to the composite group.
- ◆ `-bcv` adds only the BCVs 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.

By default, all device lists from the device group are added to the composite group.

Setting Consistency

The `[-apidb | -ppath | -rdf_consistency]` options allow you to choose a consistency method for the CG. 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 `rma11`. By default, the `remove` and `rma11` 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.

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 6-19.

Removing All Devices

When choosing to remove all devices from an existing device group, you can use the additional options to focus the operation to all devices that meet a specific criteria. 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 remove.

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. By default, this operation only affects standard devices. However, the `-vdev` option can be specified to only remove virtual devices.

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. 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
```

Renaming a Device Group

Use `symdg` 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:

```
symdg rename prod prod_B
```

Deleting a Device Group

Use `symdg` to delete a device group.

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 six 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 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.

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 via 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 the latest supported versions of Enginuity, PowerPath or other EMC products, refer to the *EMC Solutions Enabler Version 5.5 Support Matrix*.

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.

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 can be managed using the `symcgr` command.

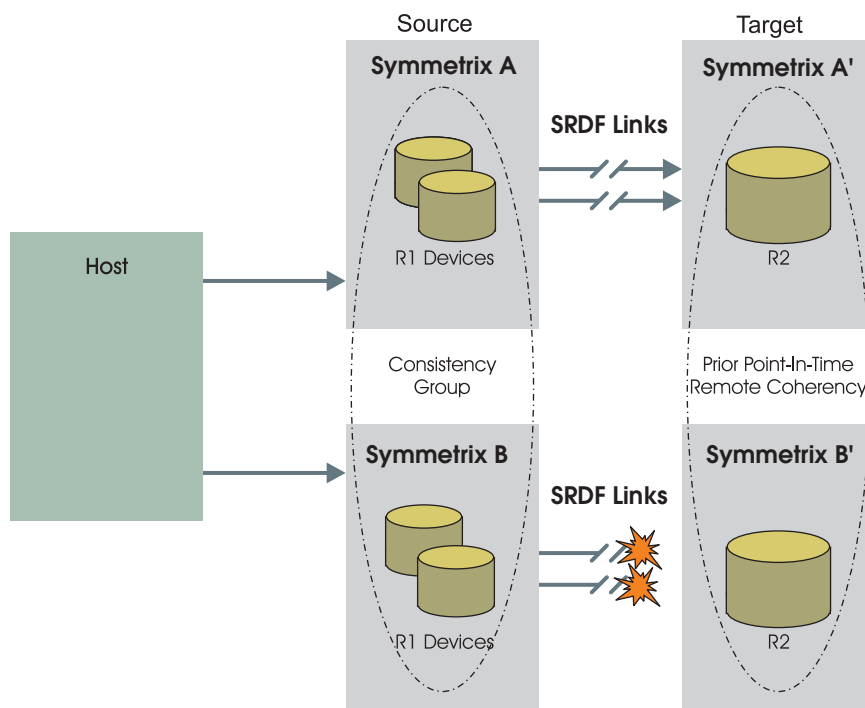


Figure 6-1 RDF Consistency Groups

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 no type is specified on a `symcg create` the group will become a REGULAR group.

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 6-25). 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 6-24). 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* on page 6-24).

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:

```
symcg -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 dskno>
```

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
V = VDEV device
R = RBCV device
Z = BRBCV device
Y = RRBCV 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 a 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]
```

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.

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 dev SymDevName [LdevName]
-cg CgName -sid SymmID
```

or

```
symcg add pd PdevName [LdevName]
-cg CgName
```

When adding devices by specifying a Symmetrix device name, the Symmetrix array's ID (-sid) is required.

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] [-R1|-R2]
[-RANGE <start:end>] [pd | dev]
[-RDFG GrpNum] [-vdev]
```

If -vdev is specified, all VDEVs for the specified Symmetrix will be added to 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]
```

or

```
symcg remove [pd] PdevName -cg CgName
[-force] [-symforce]
```

When removing devices by specifying a Symmetrix device name, the Symmetrix array's ID (*-sid*) is required.

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 -cg CgName [-sid SymmID]
[-RANGE <start:end>] [-force] [-symforce]
[-RDFG GrpNum] [-vdev] rmall
```

If *-vdev* is specified, only all VDEVs will be removed from the VDEV device list. 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.

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.

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 7-8.

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		
			Symms	RAGs	Devs	BCVs	VDEVs
MyNewCG	RDF1	Yes	1	2	32	0	0

Use the show action to return additional details about the `MyNewCG` composite group. The following shows an example output:

```
symcg show MyNewCG
```

```
Composite Group Name: MyNewCG
```

```
Composite Group Type      : RDF1
Valid                     : Yes
CG in PowerPath           : No
CG in GNS                 : No
RDF Consistency Protection Allowed : Yes
RDF Consistency Enabled   : No
```

```
Number of RDF (RA) Groups : 2
Number of STD Devices     : 32
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 (1):
```

```
{
  1) Symmetrix ID          : 000187400011
     Microcode Version     : 5671
     Number of STD Devices : 32
     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 (2):
```

```
{
  1) RDF (RA) Group Number : 27          (1A)
     Remote Symmetrix ID   : 000187400093
     Microcode Version     : 5671
```

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

```
{
```

LdevName	PdevName	Sym Dev	Device Config	Sts	Flags CSR	Cap (MB)
DEV001	/dev/rdisk/c68t12d0	0FB2	RDF1	RW	.AS	449
DEV002	/dev/rdisk/c68t12d1	0FB3	RDF1	RW	.AS	449
DEV003	/dev/rdisk/c68t12d2	0FB4	RDF1	RW	.AS	449
DEV004	/dev/rdisk/c68t12d3	0FB5	RDF1	RW	.AS	449
DEV005	/dev/rdisk/c68t12d4	0FB6	RDF1	RW	.AS	449
DEV006	/dev/rdisk/c68t12d5	0FB7	RDF1	RW	.AS	449
DEV007	/dev/rdisk/c68t12d6	0FB8	RDF1	RW	.AS	449
DEV008	/dev/rdisk/c68t12d7	0FB9	RDF1	RW	.AS	449
DEV009	/dev/rdisk/c68t13d0	0FBA	RDF1	RW	.AS	449
DEV010	/dev/rdisk/c68t13d1	0FBB	RDF1	RW	.AS	449
DEV011	/dev/rdisk/c68t13d2	0FBC	RDF1	RW	.AS	449
DEV012	/dev/rdisk/c68t13d3	0FBD	RDF1	RW	.AS	449
DEV013	/dev/rdisk/c68t13d4	0FBE	RDF1	RW	.AS	449
DEV014	/dev/rdisk/c68t13d5	0FBF	RDF1	RW	.AS	449
DEV015	/dev/rdisk/c68t13d6	0FC0	RDF1	RW	.AS	449
DEV016	/dev/rdisk/c68t13d7	0FC1	RDF1	RW	.AS	449
}						

2) RDF (RA) Group Number : 28 (1B)
 Remote Symmetrix ID : 000187400093
 Microcode Version : 5671

STD Devices (16):
 {

LdevName	PdevName	Sym Dev	Device Config	Sts	Flags CSR	Cap (MB)
DEV017	/dev/rdisk/c68t14d0	0FC2	RDF1	RW	.AS	449
DEV018	/dev/rdisk/c68t14d1	0FC3	RDF1	RW	.AS	449
DEV019	/dev/rdisk/c68t14d2	0FC4	RDF1	RW	.AS	449
DEV020	/dev/rdisk/c68t14d3	0FC5	RDF1	RW	.AS	449
DEV021	/dev/rdisk/c68t14d4	0FC6	RDF1	RW	.AS	449
DEV022	/dev/rdisk/c68t14d5	0FC7	RDF1	RW	.AS	449
DEV023	/dev/rdisk/c68t14d6	0FC8	RDF1	RW	.AS	449
DEV024	/dev/rdisk/c68t14d7	0FC9	RDF1	RW	.AS	449
DEV025	/dev/rdisk/c68t15d0	0FCA	RDF1	RW	.AS	449
DEV026	/dev/rdisk/c68t15d1	0FCB	RDF1	RW	.AS	449
DEV027	/dev/rdisk/c68t15d2	0FCC	RDF1	RW	.AS	449
DEV028	/dev/rdisk/c68t15d3	0FCD	RDF1	RW	.AS	449
DEV029	/dev/rdisk/c68t15d4	0FCE	RDF1	RW	.AS	449
DEV030	/dev/rdisk/c68t15d5	0FCF	RDF1	RW	.AS	449
DEV031	/dev/rdisk/c68t15d6	0FD0	RDF1	RW	.AS	449
DEV032	/dev/rdisk/c68t15d7	0FD1	RDF1	RW	.AS	449

```

    }
  }
}

```

Legend:

RDFA Flags:

C(onsistency) : X = Enabled, . = Disabled, - = N/A
 (RDFA) S(tatus) : A = Active, I = Inactive, - = N/A
 R(DFA Mode) : S = Single-box mode, M = MSC mode, - = N/A

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.

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] [-sid SymmID]
write_disable [LdevName [LdevName...]]
rw_enable [LdevName [LdevName...]]
ready [LdevName [LdevName...]]
not_ready [LdevName [LdevName...]]
relabel [LdevName [LdevName...]]
```

By default, this command, with no options specified, will write-disable all devices in the STD device list. If the `-bcv` option is specified, all devices in the BCV device list will be write-disabled. If the `-vdev` option is specified, all devices in the VDEV device list will be write-disabled. Specify one or a list of logical device names (`LdevName`) to perform the action on a limited number of devices within the CG.

Hold Devices in a Composite Group

Use the following syntax to hold the devices in a specified composite group:

```
symcg -cg CgName [-force]
[-sid SymmID | -rdfg SymmID:RDF_GROUP, ...]
[-bcv | -rbcv | -brbcv | -rrbcv | -vdev]
hold [LdevName [LdevName...]]
```

By default, this command, with no options specified, will hold all devices in the STD device list. You can limit the hold action to a specific Symmetrix array or a specific RDF group by specifying the Symmetrix array and RDF group number. The following options are available:

- ◆ `-bcv` holds all devices in the BCV device list.
- ◆ `-rbcv` holds all devices in the RBCV device list.
- ◆ `-brbcv` holds all devices in the BRBCV device list.
- ◆ `-rrbcv` holds all devices in the RRBCV device list.
- ◆ `-vdev` holds all devices in the VDEV device list.

Unhold Devices in a Composite Group

Use the following syntax to unhold the devices in a specified composite group:

```
symcg -cg CgName [-force]
[-sid SymmID | -rdfg SymmID:RDF_GROUP, ...]
[-bcv | -rbcv | -brbcv | -rrbcv | -vdev]
unhold [LdevName [LdevName...]]
```

By default, this command, with no options specified, will unhold all devices in the STD device list. You can limit the unhold action to a specific Symmetrix array or a specific RDF group by specifying the Symmetrix array and RDF group number. The following options are available:

- ◆ `-bcv` unholds all devices in the BCV device list.
- ◆ `-rbcv` unholds all devices in the RBCV device list.
- ◆ `-brbcv` unholds all devices in the BRBCV device list.
- ◆ `-rrbcv` unholds all devices in the RRBCV device list.
- ◆ `-vdev` unholds all devices in the VDEV device list.

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*.

Group Name Services

This chapter describes the optional Group Name Services (GNS) feature.

- ◆ Group Name Services Overview7-2
- ◆ GNS Configuration7-9
- ◆ GNS Daemon Control7-11
- ◆ How to Mirror Remote Device Group Definitions7-17
- ◆ Identify GNS Groups.....7-21
- ◆ GNS Backup and Restore Mechanism7-23
- ◆ Troubleshooting GNS.....7-25

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.

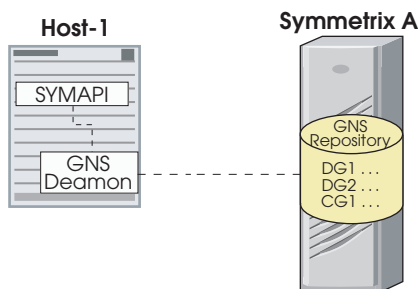


Figure 7-1 GNS Daemon Controls GNS Activity

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 on page 7-2. 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 7-13.

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).

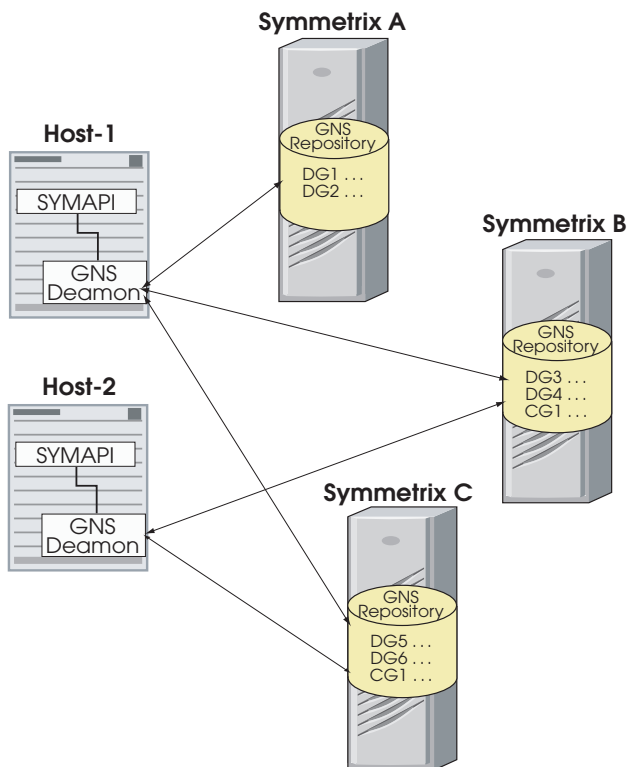


Figure 7-2 Host-Visible GNS State

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 7-6) with the updates made by Host-1. Host-3's client application will detect the change (4) on its next group call.

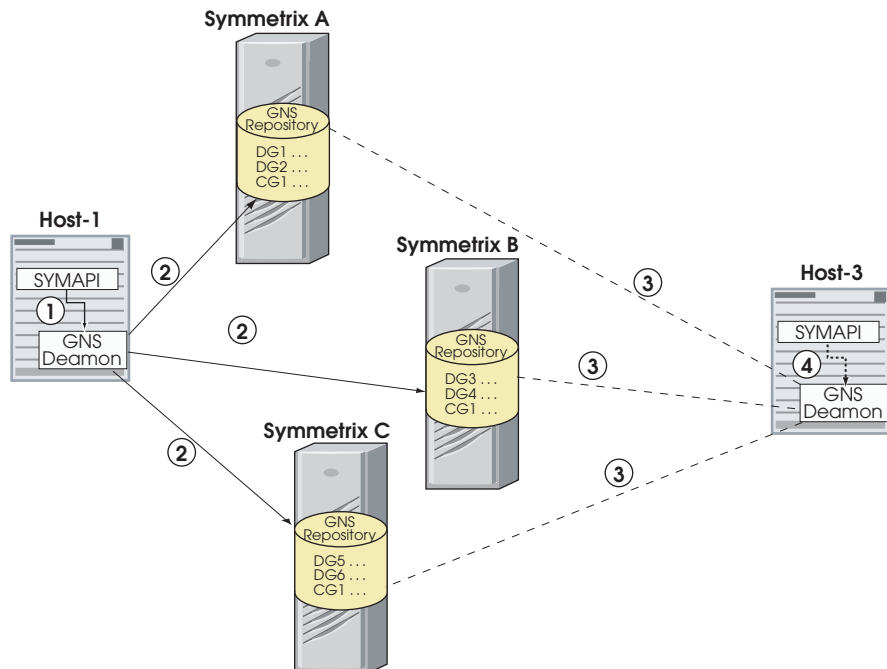


Figure 7-3 GNS Across Multiple Hosts

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 6-24. For more information on marking composite groups for MSC, refer to *Creating Composite Groups with Multi Session Consistency (MSC)* on page 6-25.

When using the `symcg create` command, the `-apidb` option can be used to override this behavior, allowing changes to only get committed to GNS.

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 7-17.

Limitation: 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 3-2 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 7-13). 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 `stord daemon` utility. These control options include:

- ◆ Stopping the daemon.
- ◆ Querying the daemon's status.
- ◆ Querying the daemon's log files.

For details, refer to the `stord daemon` 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 *stordaeomon setoption* CLI utility, which programmatically makes changes to the *daemon_options* file. The following is the command-line syntax for this option:

```
stordaeomon 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 enable or 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.

Table 7-1 GNS Daemon Optional Behavior Parameters (continued)

Optional Behavior Parameter	= <OptValue DefaultValue>	Description
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.
storgnsdLlogfile_size	= nnn 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 <i>How to Mirror Remote Device Group Definitions</i> on page 7-17.
storgnsd:gns_rmtarr_update_interval	<i>nn</i> 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 <i>How to Mirror Remote Device Group Definitions</i> on page 7-17.

Table 7-1 GNS Daemon Optional Behavior Parameters (continued)

Optional Behavior Parameter	= <OptValue DefaultValue>	Description
storgnsd:gns_shadow_file	= enable disable needed	If set to <i>enable</i> , 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 or 5x66). If set to <i>needed</i> , backups are only maintained for Symmetrix arrays with Enginuity versions lower than 5x71. Refer to <i>GNS Backup and Restore Mechanism</i> on page 7-23.
storgnsd:gns_symavoid	= <i>symid,symid,...</i>	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	= <i>nnn</i> 600	The interval, in seconds, at which to check for Symmetrix configuration changes.
storgnsd:gns_syminclude	= <i>symid,symid,...</i>	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 of both the form, 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/rdsk/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 `syndg 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 `syndg 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**.

```
Group Name: MyRegDeviceGroup

Group Type                : REGULAR
→ Device Group in GNS     : Yes
Valid                     : Yes
Symmetrix ID              : 000184500160
Group Creation Time       : Tue Dec 4 16:44:18 2004
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
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 20
Number of Remotely-assoc'd RBCV's (RBCV RDF) : 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**. 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
→ 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 5x66 or 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 7-13.

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 reconfiguration 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
```

For information on renaming composite groups, see *Renaming a Composite Group* on page 6-30.

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
◆ REQUEST Statistics.....	8-5
◆ BACKEND Statistics and PREFETCH Activity	8-10
◆ Director Statistics	8-12
◆ PORT Statistics	8-14
◆ MEMIO Statistics	8-15
◆ DMSP Stats.....	8-17
◆ Disk Statistics.....	8-18
◆ CACHE Statistics	8-19
◆ CYCLE Statistics.....	8-25

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
- ◆ Retrieves the performance counts for one or more Symmetrix devices
- ◆ 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.

Table 8-1 Performance Display Options (continued)

Option	Description
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).

REQUEST 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 010050010333, enter:

```
symstat -i 5 -c 2 -sid 333
```

Using this command, the following performance display is an example for a statistical run on a Symmetrix array:

```
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
010050010333	Local	3430	5264	2304	147	657
010050010250	Local	3430	5264	2048	147	602
010060020545	Remote	3200	5064	768	312	942
010060020549	Remote	3430	5264	3584	48	178

```
symstat -i 5 -c 2 -sid 333
```

DEVICE		IO/sec		KB/sec		% Hits		%Seq		Num WP
		READ	WRITE	READ	WRITE	RD	WRT	READ	Tracks	
14:57:26										
①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
14:57:26	012 (rrze154c)	0	1	0	1638	N/A	100	N/A		25
	013 (rrzf154c)	0	1	0	1638	N/A	100	N/A		19
	014 (rrzg154c)	0	1	0	1638	N/A	100	N/A		25
	015 (/dev/rrz155c)	0	1	0	1638	N/A	100	N/A		19
	016 (rrzb155c)	0	1	0	1638	N/A	100	N/A		25
		----	----	----	----	----	----	----	----	----
		0	5	0	6100	N/A	100	N/A		110

As shown in this example, the following statistics are returned:

1. Host time stamp of the sample.
2. Active Symmetrix device name and host physical device name.
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). This field is blank for Symm3 data.
8. Write cache hit ratio (percentage of write hits).
9. Sequential read ratio (percentage of sequential reads).
10. Write-pending track count.

Note: In this example, the performance type is not specified and, therefore, defaults to REQUESTS.

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 *REQUEST Statistics* on page 8-5, 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`), 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 `-rdfg #`, then the output display lists just the specified RDF group.
- ◆ If the `-g #` 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 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 -RepType 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.

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:

	DIRECTOR	Prefetch Trk/sec	Prefetch Used	Trk/sec Unused
16:42:30				
①	②	③	④	⑤
16:42:35	DA-2A	1	1	0
16:42:40	DA-2A	7	6	1
16:42:45	DA-2A	19	19	0
		----	----	----
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. 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 -sid 333 -dir all
```

The following display is an example of a director statistics returned:

```
symstat -i 30 -c 3 -sid 333 -dir all -type requests
```

	DIRECTOR	IO/sec	Cache READ	Requests/sec WRITE	RW	% RW Hits
16:27:31						
16:28:01	DA-2A	69	0	122	122	0
	DA-3A	34	0	60	60	0
	RA-5A	310	0	310	310	0
	DA-14A	33	0	59	59	0
	DA-2B	0	0	0	0	N/A
	DA-3B	1	0	2	2	0
	DA-14B	33	0	59	59	0
	SA-16B	12	0	254	254	100
		-----	-----	-----	-----	----
Total		492	0	866	866	29
16:28:31	DA-2A	52	0	92	92	0
	DA-3A	26	0	47	47	0
	RA-5A	311	0	311	311	0
	DA-14A	28	0	49	49	0
	DA-15A	6	0	10	10	0
	DA-2B	15	0	27	27	0
	DA-3B	14	0	25	25	0
	DA-14B	28	0	49	49	0
	SA-16B	12	0	254	255	100
		-----	-----	-----	-----	----
Total		492	0	864	865	29

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).

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 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.

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

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:

```
symdbg 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	
15:13:32	①	②	WP ③	Tracks ④	Prefchd ⑤	Destgd ⑥	WPmax
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 Stats

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, 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 333 -sa all -type cache
```

The following is an example output:

	DIRECTOR	Misses/sec Cache Read	Disconnects/sec	
			System	Device
16:39:26	2	3	4	5
16:39:26				
16:39:26				
16:39:26	SA-16A	1	0	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			Cache Requests/sec			% RW	Locks/sec
14:29:44	#	NAME	READ	WRITE	RW	HITS	
1	2	3	4	5	6	7	8
14:29:59	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`), 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 `-rdfg #`, then the output display lists just the specified RDF group.
- ◆ If the `-g #` 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: 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	RA Grp	Session				Cache Slots In Use	%Available Cache Used
		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						-----	-----
Slots						6489	4.4
GB						0.20	

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	RA Grp	Session				Cache Slots In Use	%Available Cache Used
		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						-----	-----
Slots						10955	7.5
GB						0.33	

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	RA Grp	Session				Cache Slots In Use	%Available Cache Used
		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							
Slots						7092	4.8
GB						0.22	

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 SNOW 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 SNOW device, or both. When a cache slot has both local and SNOW 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% (F0/FF) 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 SNOW 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 SNOW 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.

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 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 `-rdfg #`, then the output display lists just the specified RDF group.
- ◆ If the `-g #` 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

Device Group Name	RA Grp	Session			Cycle Time (sec)				Cycle Size	
		Type	Number	Active Cycle#	Min	Avg	Last	Switch	Active	Inactive
1	2	3	4	5	6	7	8	9	10	11
< . . . >										
RaGrpNum_05	5	RDF1	4	39086	15	16	16	9	0	0
RaGrpNum_09	9	RDF1	8	166	15	19	19	4	0	0
RaGrpNum_10	10	RDF1	9	178	15	19	19	4	0	0
RaGrpNum_11	11	RDF1	10	178	15	19	20	4	0	0
RaGrpNum_12	12	RDF1	11	177	15	19	19	4	0	0
RaGrpNum_13	13	RDF1	12	172	15	19	20	4	0	0
RaGrpNum_21	21	RDF1	20	166	15	19	19	4	0	0
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: 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).

XML Structured Output

This appendix describes the XML output option of the SYMCLI.

- ◆ XML Structured Output DataA-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

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|standard
```

Standard mode 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, 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|standard
```

The `-output` flag is not found in `-help` or `man` pages because of its wide scope and usage.

A

- Access modes
 - SYMCLI commands 3-6
- Applications registered 4-9
- Audit log
 - filtering log entries 2-13
 - symaudit 2-13
- Avoidance files 2-5, 2-6

B

- Base Daemon 3-2
- BCV devices 4-21
- Behavior parameters 2-4
- Business Continuance 4-21
- Business continuance operations 4-21

C

- Cache memory 1-4
 - LRU defined 1-4
- Cache statistics
 - LRU 8-19
- CLARiiON array
 - discover 3-2, 3-4
- Common audit log 2-13
- configuration database, *See* database

D

- DA 1-3
- database
 - access modes 3-6
 - building a new 3-2
 - changing file in use 3-6

- configuration information 3-7
- inhibiting synchronization 3-7
- location 3-5
- multi-database environment 3-5
- offline 3-6
- online 3-6
- synchronization 3-7
- device configuration 4-34
- Device groups
 - adding all ungrouped devices 6-8
 - adding devices to 6-6, 6-7
 - BCV devices in 6-5
 - creating 6-6
 - deleting 6-19
 - exporting 6-14
 - gatekeepers associated with 6-6
 - information about 6-10
 - listing 6-9
 - moving a device from 6-18
 - naming 6-4
 - RDF1 6-5
 - RDF2 6-5
 - REGULAR 6-5
 - removing a device from 6-17
 - Renaming 6-19
 - SRDF standard devices in 6-5
 - types 6-5
 - using 6-4
 - Virtual devices in 6-5
- Device lists 2-20
- device service states 4-30
- device status 4-30
- Devices
 - renaming 6-16

Directors

- configuration of 4-13
- status of 4-13

Discover

- CLARiiON 3-2, 3-4
- Symmetrix 3-2, 3-4

Disk director 1-3

Disks

- configuration of 4-37

DMSP 8-17

Dynamic mirroring service policy 8-17

Dynamic RDF 4-21

E

environment variables 2-3

Export groups to files 6-14

External locks

- device 4-35
- on Symmetrix 4-17

F

Files

- avoidance 2-5
- device import list 2-20
- log 2-8
- options 2-4
- selection 2-5

Front-end director 1-3

G

Gatekeeper devices 4-21

- associating 5-8
- candidate priority 5-2
- choosing 5-3
- database considerations of 5-5
- definition 5-2
- disassociating 5-8
- establishing 5-6
- obtain list of 5-8
- sizing 5-4
- undefining 5-7

Gatekeepers

- defined 5-4

gkavoid 2-5

gkselect 2-5

GNS

- active group list 7-8
- backup, automatic 7-23
- backup, manual 7-24
- composite groups 7-8
- daemon 7-2, 7-13, 7-14
- device groups
- output 7-21
- PowerPath interaction 7-6
- restore, manual 7-24
- shared repository
- SRDF option 7-6
- GNS, daemon options
 - Remote mirrors 7-17
 - Remote mirrors, no BCV 7-18
- Group Name Services, *See* GNS

H

HBA details 4-3

host database file, *See* database

Hosts connected 4-10

Hyper 1-4, 4-20

Hyper-volume 1-4, 4-20

I

inqfile 2-5

L

Locks

- external device 4-35
- external Symmetrix 4-17

Log file 2-8

- naming 2-9

Logging

- dated and undated file names 2-9
- enabling or disabling 2-9
- log entry date formats 2-10

Logical device name 6-4

LRU

- defined 1-4

LRU, cache management 8-19

M

Memory board information 4-11

Meta devices 4-21
 MSC 6-16, 6-25
 Multi Session Consistency 6-25

N

Network services 4-11

O

Offline
 director ports 4-19
 directors 4-19
 Online
 director ports 4-19
 directors 4-19
 Optional behavior parameters 2-4
 Options file 2-4

P

Path configuration 2-2
 Physical devices
 list configuration 4-34
 lists 4-22
 show configuration 4-34

R

RDF consistency 6-16
 Regular groups 6-5
 Remote server 2-3

S

Save devices 4-22
 SCSI INQUIRY 4-3
 Selection file 2-5, 2-6, 2-7
 Server 2-3
 service states 4-30
 SRDF devices 4-21
 Standard devices 4-21
 Statistics
 BCV devices 8-15

 cache activity 8-25
 cache memory to disk 8-3
 director 8-12
 Director Port 8-14
 I/O requests 8-3
 LRU 8-19
 performance 8-2
 prefetch activity for DAs 8-10
 prefetch disk activity 8-4
 Standard devices 8-15
 SYMCLI commands for 8-5
 statistics
 cache activity 8-3
 status of device 4-30
 Storage capacities 4-11
 storgnsd, *See* GNS, daemon
 storgnsd, *See* GNS, daemon options
 SYMAPI database, *See* database
 SYMAPI server 2-3
 symavoid 2-5
 SYMCLI commands
 symapisrv 2-3
 symcli 2-16
 syminq 4-3
 symstat 8-5
 SYMCLI Path 2-2
 Symmetrix
 device configuration 4-34
 Symmetrix array
 discover 3-2, 3-4
 Symmetrix devices
 list 4-23

V

VCM devices 4-22
 Virtual devices 4-22

X

XML output A-2

