EMC® Solutions Enabler Symmetrix® Array Management CLI

Version 7.0

Product Guide

P/N 300-002-939
REV A06

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

Published April, 2009

Contents

Chapter 8 Statistics

Chapter 9 Change Tracker

Chapter 10 Gatekeeper Devices

Chapter 11 Events and Logs

Chapter 12 XML Structured Output

Index

Figures

Tables

Preface

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

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

Audience

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

Related Documentation

Other Symmetrix publications of related interest are:

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

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

Conventions Used in This Guide

EMC uses the following conventions for special notices.

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

CAUTION

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

The following naming conventions are used in this manual:

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

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

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

Typographical Conventions

This guide uses the following type style conventions:

Palatino, bold	◆ Boldface text provides extra emphasis and emphasizes warnings, and specifies window names and menu items in text.
Palatino, italic	◆ New terms or unique word usage in text ◆ Applies emphasis in examples and in references to book titles and sections
Courier, italic	◆ Identifies variables in a software syntax (non-literal notation)

Courier	◆ A fixed space font identifies files and pathnames, and is used in command line entries, displayed text, or program listings.
	◆ System prompts and displays and specific filenames or complete paths. For example:
	`working root directory [/user/emc]:`
	`c:\Program Files\EMC\Symapi\db`

Courier, bold	◆ Actual user entry in examples. For example: **`symcfg discover`**

Where to get help

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

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

http://Powerlink.EMC.com

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

Your comments

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

techpubcomments@EMC.com

1 Introduction

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

SYMCLI introduction

The EMC® Solutions Enabler SYMCLI is a specialized library consisting of commands that can be invoked on 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 as shown in Figure 1, though some features are supported for CLARiiON® arrays as well.

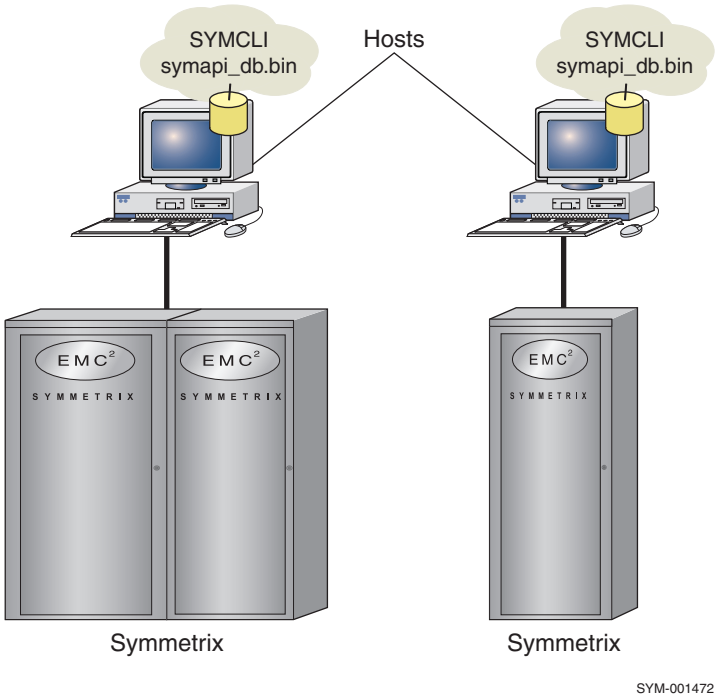


Figure 1 SYMCLI host/array interaction

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).

SYMCLI usage tips

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

Obtaining SYMCLI help

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

SYMCLI help command

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

Table 1 SYMCLI help options

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

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

Table 2 Command naming conventions

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

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

Environment variables: Altering the command line response

SYMCLI provides environment variables that can be preset to streamline and expedite your command line session. These environment variables can be set to commonly used values, which eliminates the need to specify those arguments on the command line.

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
```

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

Enabling: setenv

To set an environment variable, use the command `setenv`. For example, to always display the output of BCV and SRDF® commands in detail, set the verbose mode, `-v` option when specified on the command line, as the default display behavior as follows:

```
setenv SYMCLI_VERBOSE 1
```

Disabling: unsetenv

To disable or turn off an environment variable, use the command `unsetenv`. You can turn off or remove any environment variable. For example, to turn off the verbose mode, enter:

```
unsetenv SYMCLI_VERBOSE
```

Call the command `symcli -env` or refer to the *Solutions Enabler SYMCLI Command Reference HTML Help* for a complete list of environment variables.

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/rdsk/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.

Compatibility mode: Preserving old Solutions Enabler scripts

If you have specialized scripts that were developed against a prior version of Solutions Enabler and you are concerned that they will not work because of output changes in the latest release, you can run your commands or scripts in compatibility mode. Compatibility mode specifies the command output reporting style to be compatible with prior SYMCLI versions.

Note: While running older scripts in compatibility mode, you may not return output that is only available in higher releases.

You can set your compatibility mode as a global environment variable using `setenv SYMCLI_MODE` or at the command line using the `-mode` option on any command. Specifies the command output reporting style to be compatible with prior SYMCLI versions. Possible values are V55, V60, V61, V62, V63, V64, V65, V70.

For example, to set the compatibility mode to Solutions Enabler V6.0, enter the following on the command line:

```
symcfg list -mode V60
```

2

Discovery

This chapter describes the SYMCLI discovery process.

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 as their directors and devices, is maintained in a configuration database file on each host. Once your environment is discovered, you can direct information requests to retrieve array-level (high-level) data or device-level (low-level) information from the storage environment using SYMCLI commands.

Performing discovery

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

```
symcfg

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

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, locally attached CLARiiON arrays by specifying the `-clariion` option, only physical device information by specifying the `-pdev` option, or a specific Symmetrix array by specifying the `-sid` option (this option can also be used with the `-pdev` option to return only physical device information for a specific array). For more information about the Symmetrix configuration database, refer to "SYMAPI configuration database file" on page 28.

CLARiiON assisted discovery

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

To discover all CLARiiON arrays, local or remote, you first need to create a connectivity authorization for the targeted arrays. For more information, refer to "Connectivity authorization" on page 25.

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

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

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

```
losat246      losat247
EngDev101     EngDev112
```

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

Is the database file synchronized?

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

```
symcfg verify -sid 814

The Symmetrix configuration and the database file are in sync.

echo $status
0
```

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

```
symcfg list -status

                                S Y M M E T R I X

                                                 Mcode
    SymmID          Attachment  Model            Version  Config Changed
Discovered

    000187720758    Local       2000P-M2         5671     No             Yes
    000190100097    Local       DMX3-24          5771     No             Yes
    000000006206    Remote      2000P-M2         5671     Yes            Yes
    000187900035    Remote      800-M2           5671     No             Yes
    000187900041    Remote      800-M2           5671     Yes            Yes
    000190300174    Remote      DMX3-6           5771     Yes            Yes
    000190300175    Remote      DMX3-6           5771     No             Yes
    000190300184    Remote      DMX3-6           5771     No             Yes
```

Scanning for new devices

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

Connectivity authorization

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

```
symcfg

        authorization list [-vmware] [-v]

        authorization <add | update> -host HostName
           -username UserName -password PassWord
           [-namespace NameSpace] [-ahost HostName]
           [<-vmport | -port> port] [-vmware]

        authorization delete -host HostName -username UserName
           [-namespace NameSpace] [-vmware]
```

VMware virtual disk mapping support

Solution Enabler can now resolve all virtual disks that have only one Symmetrix device in the VMware datastore (supported with ESX 3.5 and higher). To use support VMware virtual disk mapping, credentials must be configured using the following `symcfg auth` command. For example, enter:

```
symcfg auth add -host <HostName> -username root -password <PassWord>
-namespace [vmware|esxv2] -vmware
```

`In addition,` copy the SSL certificate from the server (for example, `/etc/vmware/ssl/rui.crt`) to the VM OS as `var/symapi/config/viclient_cert.pem`.

When virtual disk mapping is configured, the `syminq` command will return the virtual disk information:

```
./syminq
      Device              Product                   Device
---------------- --------------------------- ----------------------
Name      Type   Vendor    ID           Rev  Ser Num      Cap (KB)
---------------- --------------------------- ----------------------
/dev/sda         VMware    Virtual disk 1.0  N/A          20971520
/dev/sdb         DGC       DISK         0326 0100003C      1048576
/dev/sdc         VMware    Virtual disk 1.0  N/A           8388608
/dev/sdd         EMC       SYMMETRIX    5773 8600022000      92160
/dev/sde         VMware    Virtual disk 1.0  N/A           8388608
/dev/dm-0        VMware    Virtual disk 1.0  N/A           8388608
```

Note: An expected capacity difference is visible between the output of `syminq` and `sympd list`. When creating the virtual disk, it can use all the space or less space than the Symmetrix device; so the `syminq` output may show smaller capacity than what `sympd` shows. In some cases, `syminq` will show larger capacity than `sympd`, because the vmware datastore (or virtual disk) can have multiple Symmetrix devices.

The `sympd` will display virtual disk as a Symmetrix devices. The example below shows an ouput of `sympd list`.:

```
sympd list
Symmetrix ID: 000190300186
Device Name           Directors                  Device
--------------------------------------------------------------------------
Cap
Physical               Sym  SA :P DA :IT  Config        Attribute    Sts   (MB)
-------------------------- ------------ ------------------------------------
/dev/sdd               0022 16C:0 01A:C6  2-Way Mir     N/Grp'd  VCM RW      90
/dev/sde               0085 16C:0 02B:D5  2-Way Mir     N/Grp'd      RW    8631
```

Synchronizing configuration updates with the database

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

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

Sync operations

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

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

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

Table 3 Database synchronization options

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

Note: The Solutions Enabler V7.0 configuration database format has changed. Solutions Enabler V7.0 can read database files written by an older version of Solutions Enabler, but earlier versions of Solutions Enabler cannot read a configuratoin database written by Solutions Enabler V7.0.

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

Note: For information about discovering your storage environment and building your Symmetrix configuration database, refer to "Discovering configuration and status data" on page 22.

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.

database file lock

Solutions Enabler V7.0 utilizes a configuration database locking file. The lock file is creates automatically and is given has the same name as the configuration database file, but it is appended with a *_xlock* suffix. For example, symapi_db.bin_xlock.

Solutions Enabler uses this lock file to serialize access to the configuration database file. It contains no data and is only used as a lock.

If you protect the symapi_db.bin file to restrict the users permitted to perform Solutions Enabler management operations, this lock file should be protected as well. Both symapi_db.bin and symapi_db.bin_xlock should be given the same level of protection.

Changing the current database file

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

```
symcli -def
```

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

For example, to change the database file to `symbackup_db.bin` from a UNIX host in C shell (csh), enter:

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

To perform the same operation on Windows, enter:

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

Changing the database in client/server mode

For security reasons, in client/server mode the configuration database file must be located within the default database directory.

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

```
/var/symapi/db
```

On Windows, the default configuration database path is:

```
C:\Program Files\EMC\Symapi\db
```

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) from a UNIX host in C shell (csh), enter:

```
setenv SYMCLI_OFFLINE 1
```

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

Database configuration information

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

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

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

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

GNS State                                        : Disabled
```

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

3

Configuration Data

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

Configuration data: Layer by layer

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

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

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

SCSI inquiries

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

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

The `syminq` command 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.

The returned results can also be limited to devices with target mapping information (only for devices mapped through fibre HBAs using the `-mapinfo` option or, for Symmetrix or CLARiiON device, to devices given device identifiers by a user or application by using the `-identifier` option; identifiers are limitedto `device_name`, `nice_name`, `hp_id`, and `vms_id`.

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

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

Example: SCSI device information

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

```
syminq
```

```
      Device              Product                    Device
---------------- ---------------------------- ----------------------
Name       Type  Vendor    ID           Rev  Ser Num       Cap (KB)
---------------- ---------------------------- ----------------------
/dev/sda         MAXTOR    ATLAS10K4_3* DFL0 B2CS5FSM      35916548
/dev/sdb   GK    EMC       SYMMETRIX    5772 150009E020         5760
/dev/sdi   GK    EMC       SYMMETRIX    5772 15000A7020         5760
/dev/sdj   GK    EMC       SYMMETRIX    5772 15000A8020         5760
/dev/sdk   GK    EMC       SYMMETRIX    5772 15000A9020         5760
/dev/sdl   GK    EMC       SYMMETRIX    5772 15000AA020         5760
/dev/sdm   GK    EMC       SYMMETRIX    5772 15000AB020         5760
/dev/sdn   GK    EMC       SYMMETRIX    5874 0700022000         5760
/dev/sdo   GK    EMC       SYMMETRIX    5874 0700023000         5760
/dev/sdp   GK    EMC       SYMMETRIX    5874 0700024000         5760
/dev/sdq   GK    EMC       SYMMETRIX    5874 0700025000         5760
/dev/sdr   GK    EMC       SYMMETRIX    5874 0700026000         5760
/dev/sds   GK    EMC       SYMMETRIX    5874 0700027000         5760
/dev/sdt   R1    EMC       SYMMETRIX    5874 0700042000       919680
/dev/sdu   R1    EMC       SYMMETRIX    5874 0700043000       919680
/dev/sdv   R1    EMC       SYMMETRIX    5874 0700044000       919680
/dev/sdw   R1    EMC       SYMMETRIX    5874 0700045000       919680
/dev/sdx   R1    EMC       SYMMETRIX    5874 0700046000       919680
/dev/sdy   R1    EMC       SYMMETRIX    5874 0700047000       919680
```

```
/dev/sdz  R2     EMC       SYMMETRIX    5874 0700048000     919680
/dev/sdaa R2     EMC       SYMMETRIX    5874 0700049000     919680
/dev/sdch CKD    EMC       SYMMETRIX    5772 5500228000     934920
/dev/sdci CKD    EMC       SYMMETRIX    5772 5500229000     934920
/dev/sdcj CKD    EMC       SYMMETRIX    5772 550022A000     934920
```

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/sdb  GK     000190300215     5772 150009E020       5760
/dev/sdc  GK     000190300215     5772 150009F020       5760
/dev/sdd  GK     000190300215     5772 15000A2020       5760
/dev/sde  GK     000190300215     5772 15000A3020       5760
/dev/sdf  GK     000190300215     5772 15000A4020       5760
/dev/sdg  GK     000190300215     5772 15000A5020       5760
/dev/sdh  GK     000190300215     5772 15000A6020       5760
/dev/sdi  GK     000190300215     5772 15000A7020       5760
/dev/sdj  GK     000190300215     5772 15000A8020       5760
/dev/sdk  GK     000190300215     5772 15000A9020       5760
/dev/sdl  GK     000190300215     5772 15000AA020       5760
/dev/sdm  GK     000190300215     5772 15000AB020       5760
```

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/rdsk/c5t0d0s2        APM00034501433   0216 09000017      2097152
/dev/rdsk/c5t0d1s2        APM00034501433   0216 07000017      2097152
```

Example: Devices without capacity

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

syminq -symmids -nocap

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

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

```
syminq -cids -nocap
```

```
        Device                  Clariion          Device
------------------------ -------------------- ----------
        Name                ID                Rev  Ser Num
------------------------ -------------------- ----------

/dev/rdsk/c5t0d0s2       APM00034501433    0216 09000017
/dev/rdsk/c5t0d1s2       APM00034501433    0216 07000017
```

Example: Devices with WWN

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

```
syminq -sym -wwn
```

```
Device                                          Device
------------------------------ ---------------- --------------------------------
Name                        Num  Array ID         WWN
------------------------------ ---------------- --------------------------------
/dev/sdv                    0028 000190300016     60060480000190300016533030303238
/dev/sdw                    0029 000190300016     60060480000190300016533030303239
/dev/sdx                    002A 000190300016     60060480000190300016533030303241
/dev/sdy                    002B 000190300016     60060480000190300016533030303242
/dev/sdz                    002C 000190300016     60060480000190300016533030303243
```

Example: Colon formatted WWN

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

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

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

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/rdsk/c2t0d0s2 0017 15D  0
000187900771 /dev/rdsk/c2t0d1s2 0018 15D  0
000187900771 /dev/rdsk/c2t0d2s2 0019 15D  0
000187900771 /dev/rdsk/c2t0d3s2 001A 15D  0
```

HBA information

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

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

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

Example: HBA inquiries

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

`syminq hba`

```
  Host Name : api171

  HBA Type            : FibreChannel
  HBA Name            : Emulex-LPe11002-E-1
  Vendor              : Emulex Corporation
  Model               : LPe11002-E
  Serial Number       : VM63487963
  Firmware Version    : 2.50A4 (Z2F2.50A4)
  Driver Version      : 8.1.10.3; HBAAPI(I) v2.1.d, 07-28-06
  Node WWN            : 0000000000000000
  Number of Ports     : 2

      Port WWN            : 10000000c9594dce
      Port name           : /sys/class/scsi_host/host1
      Port type           : NPort
      Port FCID           : 7434496
      Port speed          : 4gbit
      Supported speed     : 4gbit
      Port state          : Online
      Supported COS       : 00000008
      Supported FC4 types : 00000100000000010000000000000000000000000000000000000000000000000000
      Active FC4 types    : 00000100000000010000000000000000000000000000000000000000000000000000
      Max frame size      : 2048

      Port WWN            : 10000000c9594dcf
      Port name           : /sys/class/scsi_host/host2
      Port type           : Unknown
      Port FCID           : 0
      Port speed          : Unknown
      Supported speed     : 4gbit
      Port state          : LinkDown
      Supported COS       : 00000008
      Supported FC4 types : 00000100000000010000000000000000000000000000000000000000000000000000
      Active FC4 types    : 00000100000000010000000000000000000000000000000000000000000000000000
      Max frame size      : 2048
```

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, the Port state, the supported classes of service, supported FC4 types, active FC4 types, and max frame size shall be output for all Fibre Channel HBA ports.

Note: HBA returned data will vary slightly when the `-scsi` or `-iscsi` option is specified.

Mapping information

The `syminq` command returns target mapping information when the `-mapinfo` option is specified. The following is the `syminq -mapinfo` syntax:

```
syminq -mapinfo[PdevName]
    [-sym[-powerpath]|-clariion[-powerpath]|-hds|-storworks]
    [-cache | -nocache][-colons][-winvol]
```

When the `syminq -mapinfo` command is called, the physical device name, HBA port WWN, and target port WWN is output for each host device. The following is an example of the `syminq -mapinfo` command output:

`syminq -mapinfo`

```
               Device                          Target Mapping
----------------------------------- ---------------------------------
Name                                HBA Port WWN     Target Port WWN
----------------------------------- ---------------------------------

/dev/rdsk/c0t0d0s2                  N/A              N/A
/dev/rdsk/c1t1d0s2                  N/A              N/A
/dev/rdsk/c2t5000097208013558d0s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d1s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d2s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d3s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d4s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d5s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d6s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d7s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d8s2   210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d9s2   210000e08b80a234 5000097208013558
```

Device identifiers

To list the Symmetrix device identifiers assigned to devices by the user or other applications, use the `syminq -identifier` option and specify one of the four options: `device_name`, `nice_name`, `hp_id`, or `vms_id`.

Example: -identifier device_name

`syminq -identifier device_name`

```
      Device                                    Device
-------------- ----------------------- ----------------
Name         Num Vendor          Array ID Name
-------------- ----------------------- ----------------
/dev/sdg    0020 EMC         000190300343 test device_name
/dev/sdh    002B EMC         000190300343 N/A
/dev/sdi    002C EMC         000190300343 N/A
/dev/sdj    002D EMC         000190300343 N/A
```

Example: -identifier nice_name

`syminq -identifier nice_name`

```
      Device                                    Device
-------------- ----------------------- ----------------
Name         Num Vendor          Array ID Nice Name
-------------- ----------------------- ----------------
/dev/sdb   00200 DGC          HK192200881 LUN 200
/dev/sdc   00201 DGC          HK192200881 LUN 201
/dev/sdd   00202 DGC          HK192200881 LUN 202
/dev/sdg    0020 EMC         000190300343 0001903003430020
/dev/sdh    002B EMC         000190300343 000190300343002B
/dev/sdi    002C EMC         000190300343 000190300343002C
/dev/sdj    002D EMC         000190300343 000190300343002D
```

Example: -identifier hp_id

`syminq -identifier hp_id`

```
      Device                                    Device
-------------- ----------------------- ----------------
Name         Num Vendor          Array ID HP ID
```

```
--------------- ----------------------- ----------------
/dev/sdg    0020 EMC         000190300343 N/A
/dev/sdh    002B EMC         000190300343 test hp_id
```

Example: -identifier hp_id

syminq -identifier vms_id

```
     Device                                    Device
--------------- ----------------------- ----------------
Name         Num Vendor          Array ID VMS ID
--------------- ----------------------- ----------------

/dev/sdg    0020 EMC         000190300343 32
/dev/sdi    002C EMC         000190300343 44
/dev/sdj    002D EMC         000190300343 45
```

Storage array data

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

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

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

List arrays

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

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

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

```
symcfg list

                                S Y M M E T R I X

                                          Mcode    Cache      Num Phys  Num Symm
    SymmID       Attachment  Model        Version  Size (MB)  Devices   Devices

    000192600207 Local       Blue         5874       12288        20      1742
    000190100097 Local       DMX3-24      5771       98304        20     32088
```

```
000000006206 Remote      2000P-M2  5671       32768          0      4139
000187900035 Remote      800-M2    5671        8192          0      1314
000187900041 Remote      800-M2    5671        8192          0       960
000190300174 Remote      DMX3-6    5771       16384          0       335
000190300175 Remote      DMX3-6    5771       16384          0       914
000190300184 Remote      DMX3-6    5771       16384          0       417
```

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

```
symcfg list -clariion
```

Symmetrix configuration data

To obtain detailed configuration and status data for the Symmetrix array, enter:

```
symcfg list -sid 0207 -v

Symmetrix ID: 000192600207
Time Zone   : EST

    Product Model                          : Blue
    Symmetrix ID                           : 000192600207

    Microcode Version (Number)             : 5874 (16F20000)
    Microcode Date                         : 12.16.2008

    Microcode Patch Date                   : 00.00.2000
    Microcode Patch Level                  : 63

    Cache Size (Mirrored)                  :   12288 (MB)
    # of Available Cache Slots             :  161616
    # of PermaCache Slots In Use           :       2
    Max # of System Write Pending Slots    :  129419
    Max # of DA Write Pending Slots        :       0
    Max # of Device Write Pending Slots    :    6470

    Symmetrix Total Operating Time         : 1107 days, 19:49:50
    Symmetrix Power ON Time                : Sat Dec 17 20:55:16 2005
    Symmetrix Last IPL Time (Cold)         : Sun Dec 18 02:31:00 2005
    Symmetrix Last Fast IPL Time (Hot)     : Thu Jan  1 00:00:00 2004

    Host DB Sync Time                      : Mon Dec 29 16:45:06 2008
    Symmetrix CLI (SYMCLI) Version         : X7.0.0.151 (Edit Level: 892)
    Built with SYMAPI Version              : X7.0.0.151 (Edit Level: 892)
    SYMAPI Run Time Version                : X7.0.0.151 (Edit Level: 892)
    SYMAPI Server Version                  : unavailable

    Number of Configured (Sym) Devices     : 742
    Number of Visible (Host) Devices       :  20
    Number of Configured Actual Disks      :  32
    Number of Configured Hot Spares        :   0
    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       : A22FFFEB
    Switched RDF Configuration State       : Enabled
```

```
Concurrent RDF Configuration State   : Enabled
Dynamic RDF Configuration State      : Enabled
Concurrent Dynamic RDF Configuration : Enabled
RDF Data Mobility Configuration State: Disabled
Access Control Configuration State   : Enabled
Device Masking (VCM) Config State    : Mixed
VCMdb Access Restricted State        : N/A
Multi LRU Device Assignment          : NONE
Disk Group Assignments               : Not in Use
Hot Swap Policy                      : Permanent
Symmetrix Disk Library               : Disabled
FBA Geometry Emulation               : Native
3 Dynamic Mirrors                    : Enabled
Cache Partitioning                   : Disabled
IPSec Status                         : Pass Thru
Allow spare in mirror 4 position     : Disabled

Parity Raid Configuration            : N/A
Raid-5 Configuration                 : RAID-5 (3+1 and 7+1)
Raid-6 Configuration                 : N/A
PAV Mode                             : DynamicStandardPAV
PAV Alias Limit                      : 31

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

Auto Meta                            : Enabled
Minimum Auto Meta Size               : 131334
Auto Meta Member Size                : 65520
Auto Meta Configuration              : Concatenated
```

Note: Control operations are not allowed on devices that are on a Symmetrix Disk Library (SDL). The following SYMCLI commands will block operations on devices that are on an SDL: `symrdf`, `symrcopy`, `symmir`, `symsnap`, `symclone`, `symdev`, and `symconfigure`.

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 array `0207`, enter:

```
symcfg list -applications -sid 0207

Symmetrix ID     : 000192600207

            Host                          Application
----------------------------  ----------------------------------------------
Node Name    IP Address       ID              Vendor ID        Version   Attr
-----------  ---------------  --------------- ---------------- --------------

host001      N/A              SYMCLI          EMC Corp         7.0.0.151   -
                              SYMCFG          EMC Corp         7.0.0.151   -
```

Hosts connections to Symmetrix arrays

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

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

```
    Host               Symmetrix                    Capacity (GB)
------------  --------------------------  ------------------------------------
Node Name     ID           Director Port  Mapped STDs Mapped BCVs Paired BCVs
------------  ------------ -------- ----  ----------- ----------- -----------

6I            000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
6I totals:                                      130.8        54.8         8.4

LBQA0074      000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
LBQA0074 totals:                                  0.0         0.0         0.0

api105        000190100097 FA-3A     0         134.9         0.0         0.0
                                                ----------- ----------- -----------
api105 totals:                                  134.9         0.0         0.0

api150        000190100097 FA-3A     0         134.9         0.0         0.0
                                                ----------- ----------- -----------
api150 totals:                                  134.9         0.0         0.0

api31         000190100097 FA-7A     0         404.6         0.0         0.0
                                                ----------- ----------- -----------
api31 totals:                                   404.6         0.0         0.0

lbqa0074      000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
lbqa0074 totals:                                  0.0         0.0         0.0
```

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

If you had IPv6 addresses or host names, which would appear truncated in the default display layout, you could specify the `-ipv6` option to specify that the layout should use a display layout that does not truncate node names or addresses.

To list all the host connections, sorted by host names as shown below, of Symmetrix arrays to which your host is connected, enter:

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

```
Host                                    Symmetrix
--------------------------------------------------  --------------------------
Node Name    IP Address      OS Name  OS Revision  ID           Director Port
------------ --------------- -------- -----------  ------------ -------- ----

6I           172.23.195.65   WinNT    5.2.3790     000190100097 FA-3B     0

LBQA0074     172.23.195.74   WinNT    5.2.3790     000190100097 FA-3B     0

api105       172.23.192.105  SunOS    5.8          000190100097 FA-3A     0

api150       172.23.192.150  WinNT    5.2.3790     000190100097 FA-3A     0

api31        172.23.192.31   SunOS    5.8          000190100097 FA-7A     0
```

Viewing storage capacities to hosts

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

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

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

```
    Host               Symmetrix                  Capacity (GB)
------------  -------------------------  ----------------------------------
Node Name     ID           Director Port  Mapped STDs Mapped BCVs Paired BCVs
------------  ------------ -------- ----  ----------- ----------- -----------

6I            000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
6I totals:                                      130.8        54.8         8.4

LBQA0074      000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
LBQA0074 totals:                                  0.0         0.0         0.0

api105        000190100097 FA-3A     0         134.9         0.0         0.0
                                                ----------- ----------- -----------
api105 totals:                                  134.9         0.0         0.0

api150        000190100097 FA-3A     0         134.9         0.0         0.0
                                                ----------- ----------- -----------
api150 totals:                                  134.9         0.0         0.0

api31         000190100097 FA-7A     0         404.6         0.0         0.0
                                                ----------- ----------- -----------
api31 totals:                                   404.6         0.0         0.0

lbqa0074      000190100097 FA-3B     0         135.0        67.4         8.4
                                                ----------- ----------- -----------
lbqa0074 totals:                                  0.0         0.0         0.0
```

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

Director configuration data

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

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

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

```
symcfg -dir all list -sid 097

Symmetrix ID: 000190100097

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

    Ident  Symbolic  Numeric  Slot  Type          Status

    DF-1A     01A       1       1   DISK          Online
    DF-2A     02A       2       2   DISK          Online
    FA-3A     03A       3       3   FibreChannel  Online
    FA-4A     04A       4       4   FibreChannel  Online
    DF-5A     05A       5       5   DISK          Online
    DF-6A     06A       6       6   DISK          Online
    FA-7A     07A       7       7   FibreChannel  Online
    FA-10A    10A      10      10   FibreChannel  Online
    DF-11A    11A      11      11   DISK          Online

<. . .>
```

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 -sid 039

Symmetrix ID: 000187900039

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

    Ident  Symbolic  Numeric  Slot  Type          Status

    FA-2C     02C      34       2   FibreChannel  Online
    FA-15C    15C      47      15   FibreChannel  Online
    FA-16C    16C      48      16   FibreChannel  Online
    FA-2D     02D      50       2   FibreChannel  Online
    FA-15D    15D      63      15   FibreChannel  Online
    FA-16D    16D      64      16   FibreChannel  Online
```

Director port data

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

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

Symmetrix ID: 000190100097

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


                          Director   Port Status     Connection Status
  Ident  Type             Status     P0  P1  P2  P3    P0  P1  P2  P3

  FA-3A  FibreChannel     Online     ON  ON  N/A N/A    X   -   -   -
  FA-4A  FibreChannel     Online     ON  ON  N/A N/A    -   -   -   -
  FA-7A  FibreChannel     Online     ON  ON  N/A N/A    X   -   -   -
  FA-10A FibreChannel     Online     ON  ON  N/A N/A    -   -   -   -
  FA-13A FibreChannel     Online     ON  ON  N/A N/A    -   -   -   -
  FA-14A FibreChannel     Online     ON  ON  N/A N/A    -   -   -   -
  FA-3B  FibreChannel     Online     ON  ON  N/A N/A    X   -   -   -
  < . . .>
```

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

Addresses of devices mapped to directors

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

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

Symmetrix ID: 000190100097

          Director                  Device Name          Attr     Address
    ----------------------- ------------------------------ ---- -------------
    Ident   Symbolic  Port Sym   Physical                     VBUS  TID  LUN
    ------  --------  ---- ----  -----------------------      ----  ---  ---

    FA-3A   03A       0    0E2A  /dev/vx/rdmp/c5t0d0s2           0   00  000
                           0E2B  /dev/vx/rdmp/c5t0d1s2           0   00  001
                           04AA  /dev/vx/rdmp/c5t0d2s2   (M)     0   00  002
                           04AE  /dev/vx/rdmp/c5t0d3s2   (M)     0   00  003
                           04B2  /dev/vx/rdmp/c5t0d4s2   (M)     0   00  004
                           04B6  /dev/vx/rdmp/c5t0d5s2   (M)     0   00  005
                           04BA  /dev/vx/rdmp/c5t0d6s2   (M)     0   00  006
                           04BE  /dev/vx/rdmp/c5t0d7s2   (M)     0   00  007
                           04C2  /dev/vx/rdmp/c5t0d8s2   (M)     0   00  008
                           04C6  /dev/vx/rdmp/c5t0d9s2   (M)     0   00  009

    FA-7A   07A       0    0E2C  Not Visible                    0   00  000
                           0E2D  Not Visible                    0   00  001
                           046A  Not Visible                    0   00  002
                           046B  Not Visible                    0   00  003
                           046C  Not Visible                    0   00  004
  < . . .>
```

Next available device address

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

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

Symmetrix ID: 000190100097

          Director                 Device Name          Attr     Address
    ---------------------- ------------------------------ ---- -------------
    Ident   Symbolic  Port Sym   Physical                     VBUS  TID  LUN
    ------  --------  ---- ----  -----------------------      ----  ---  ---

    FA-3A   03A       0    0E2A  /dev/vx/rdmp/c5t0d0s2           0   00  000
                           0E2B  /dev/vx/rdmp/c5t0d1s2           0   00  001
                           04AA  /dev/vx/rdmp/c5t0d2s2   (M)     0   00  002
                           04AE  /dev/vx/rdmp/c5t0d3s2   (M)     0   00  003
                           04B2  /dev/vx/rdmp/c5t0d4s2   (M)     0   00  004
                           04B6  /dev/vx/rdmp/c5t0d5s2   (M)     0   00  005
                           04BA  /dev/vx/rdmp/c5t0d6s2   (M)     0   00  006
                           04BE  /dev/vx/rdmp/c5t0d7s2   (M)     0   00  007
                           04C2  /dev/vx/rdmp/c5t0d8s2   (M)     0   00  008
                           04C6  /dev/vx/rdmp/c5t0d9s2   (M)     0   00  009
                           -     AVAILABLE                      0   00  00A *
   <. . .>
```

Taking RA directors offline

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

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

To bring the director back online, enter:

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

Taking front-end director ports offline

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

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

CAUTION

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

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

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

Symmetrix external locks

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

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

All possible locks

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

```
symcfg list -lockn all
```

The following is an example output:

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

                                 Lock       Lock    Lock                Time
    SymmID       Attachment      Status     Number  Usage               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 previous example, the returned list contains three local Symmetrix arrays that have no known locks, as specified by the N/A values. However, remote Symmetrix 000187900039 has an exclusive lock number 15 for a configuration change activity.

Details about a specific lock number

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

```
symcfg list -lockn 0
```

```
                          S Y M M E T R I X      L O C K S
                                Lock       Lock     Lock                Time
SymmID          Attachment    Status     Number   Usage               Held (Sec)

000000006196 Local         Unknown    N/A      N/A                 N/A
000184600063 Local         Unknown    N/A      N/A                 N/A
000184600282 Local         Unknown    N/A      N/A                 N/A
000187900039 Remote        Unknown    N/A      N/A                 N/A
```

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

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

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

Details about a specific lock

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

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

```
Symmetrix ID: 000000006190

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

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

In addition, the verbose option (`-v`) can be used to provide, if available from the base daemon, extended lock information about the application and host that owns the lock. Note the 3 new fields, highlighted on bold, added tot he verbose output with Solutions Enabler V7.0: Lock Owner (identifies which application owns the lock, either Host Application, Mainframe Host Application, Config Server, Internal, Held By EMC Customer Engineer), Config Change Session ID, Migration Session Name.

For example:

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

```
symcfg list -sid 207 -lockn 15 -v

Symmetrix ID: 000192600207
```

```
Symmetrix ID                          : 000192600207   (local)
Lock Number                           : 15
Lock Usage                            : Config Change
Time Held in Seconds                  : 1801
Lock Holder ID                        : 0xf000128e

Initiator                             : 0
Director Number                       : 55
Logical Path ID                       : -1
Lock Owner                            : Config Server
Config Change Session ID              : 4356
Migration Session Name                : N/A
```

Release a lock

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

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

Differences releasing locks with SE V7.0: Enginuity 57xx vs. 5874

Versions of Solutions Enabler earlier than V7.0 allows users to release lock 15 on a DMX-4 using the `-force` option. For example

```
symcfg release -sid <xxx> -lockn 15 -force
```

With Enginuity 57*xx*, lock 15 can have only two potential owners: an RDF application when performing RDF controls or the configuration server to guarantee exclusivity when performing changes.

With versions of Solutions Enabler earlier than V7.0, the `symcfg release` command determines the lock owner and releases the lock if held by an RDF control or attempts to abort the release if held by a configuration change session.

This behavior will remain the same with Solutions Enabler 7.0 when executing the command on an array running Enginuity 57xx, but will change when executing the command on an array running Enginuity 5874.

On an array running Enginuity 5874, if lock 15 is held by the configuration server, the SYMCLI will not allow the user to abort the configuration change session. Instead, the command will fail with a message that identifies the lock owner. Users can then use `symconfigure abort` command to abort the session.

So, when Solutions Enabler 7.0 is run against an array running Enginuity 5874, the `symcfg release` command determines the lock owner and releases the lock if held by an RDF control operation, or if held by the configuration change server, the SYMCLI will block the user command and print a failure message that identifies the lock owner. The lock owner can be a configuration change session, a migrate session, or an internal change. The user can then abort the correct session that owns the lock, using the `symconfigure` or the `symmigrate` command, whichever is applicable.

Example: Solutions Enabler V7.0 and Enginuity 5874 release lock behavior

The following examples shows three instances of the `symcfg release` command on lock 15:

```
symcfg release -sid 343 -lockn 15 -force

Release lock #15 for Symmetrix unit 000190300343 (y/[n]) ?  y

Lock 15 is held by a configuration change session <654>.
The session should be aborted to release the lock.
```

```
symcfg release -sid 343 -lockn 15 -force

Release lock #15 for Symmetrix unit 000190300343 (y/[n]) ?  y

Lock 15 is held by a migration session: <session name>
The session should be aborted to release the lock.

symcfg release -sid 343 -lockn 15 -force

Release lock #15 for Symmetrix unit 000190300343 (y/[n]) ?  y

Lock 15 is held by an internal config change session and cannot be released.
```

For information about device external locks that only target specific devices, refer to "Releasing device external locks" on page 85.

SYMAPI semaphores

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

```
symcfg list -semaphores

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

                                Lock Proc.
 ID     Key        State     Type Wait  Lock Full Path Specification
 ------ ---------- --------- ---- ----- ----------------------------

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

LRU cache management group

All, or a specific, least-recently-used (LRU) cache management group(s) for one or all Symmetrix array(s) can be obtained using 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 the bottom.

Viewing network services

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

```
symcfg list -services
```

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

Viewing memory board information

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

```
symcfg list -memory
```

Mainframe CU image

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

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

```
symcfg list -cuimage
```

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

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

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

Enginuity patches

To list all of the Enginuity patches installed on a specific Symmetrix array, enter:

```
symcfg list -upatches -sid 0207

Symmetrix ID: 000192600207

 Enginuity Level    : 5874_63
 Total Patches      : 2371
 Permanent Patches  : 2347
 Temporary Patches  : 24

 P00022136:001 P00033135:001 P00033563:012 P00034999:002 P00035237:001
 P00035422:001 P00035530:009 P00036633:001 P00036812:001 P00036889:002
 P00036896:001 P00037223:004 P00037353:001 P00037388:001 P00037536:001
 P00037577:001 P00037580:001 P00037648:001 P00037747:001 P00037817:002
 P00037894:001 P00038171:001 P00038219:001 P00038287:001 P00038326:001
 P00038347:001 P00038352:001 P00038403:001 P00038485:001 P00038612:006
 P00038713:001 P00038785:001 P00038821:001 P00038896:004 P00038921:007
 P00038967:001 P00039002:002 P00039013:001 P00039089:001 P00039111:002
 T00039113:002 P00039146:001 P00039151:001 P00039195:001 P00039208:001
 P00039224:001 P00039289:001 P00039300:001 P00039319:001 P00039338:001
. . .
```

Environmental data

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

```
symcfg -sid 150 list -env_data
```

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

```
symcfg -sid 150 show -env_data SystemBay
```

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

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

```
symcfg -sid 150 list -env_data -service_state failed
```

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

Example: Arrays with system bay and drive bays

The following example lists environmental output for Symmetrix arrays running Enginuity 5874 or higher containing a system bay and at least one drive bay:

```
symcfg -sid 150 list -env_data

Symmetrix ID               : 000192600150
Timestamp of Status Data   : 02/03/2009 12:06:42

    System Bay

      Bay Name                              :    SB-1
      Number of Standby Power Supplies      :    2
      Number of Drive Enclosures            :    0
      Number of Enclosure Slots             :    1
      Number of MIBE Enclosures             :    2

      Summary Status of Contained Modules
        All Standby Power Supplies          :    Normal
        All Enclosure Slots                 :    Normal
        All Power Supplies                  :    Normal
        All Fans                            :    Normal
        All Management Modules              :    Normal
        All IO Module Carriers              :    Normal
        All Directors                       :    Normal
        All MIBE Enclosures                 :    Normal
        All Power Supplies                  :    Normal

    Drive Bays

      Bay Name                              :    DB-1A
      Number of Standby of Power Supplies   :    2
      Number of Drive Enclosures            :    8

      Summary Status of Contained Modules
        All Enclosures                      :    Normal
```

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

The following example shows detailed system bay output for Symmetrix arrays running Enginuity 5874 or higher containing a system bay and at least one drive bay:

```
symcfg show -sid 150 SB-1 -env_data

Symmetrix ID               : 000192600150
Timestamp of Status Data   : 02/03/2009 12:06:42

    System Bay

      Bay Name                              :    SB-1
      Number of Standby Power Supplies      :    0
      Number of Drive Enclosures            :    0
      Number of Enclosure Slots             :    1
      Number of MIBE Enclosures             :    2

      Status of Contained Modules
        Standby Power Supplies

        Enclosure Slot Number               :    4
          Enclosure Slot State              :    Normal
          PS-A                              :    Normal
          PS-B                              :    Normal
          FAN-A                             :    Normal
          FAN-B                             :    Normal
          FAN-C                             :    Normal
          FAN-D                             :    Normal
          MM-A                              :    Normal
          MM-B                              :    Normal
          IOMC-A                            :    Normal
          IOMC-B                            :    Normal
          DIR-7                             :    Normal
          DIR-8                             :    Normal

        MIBE Name                           :    MIBE-L-2A
          MIBE State                        :    Normal
          PS-A                              :    Normal
          PS-B                              :    Normal

        MIBE Name                           :    MIBE-L-2B
          MIBE State                        :    Normal
          PS-A                              :    Normal
          PS-B                              :    Normal
```

The following example shows detailed drive bay output for Symmetrix arrays running Enginuity 5874 or higher containing a system bay and at least one drive bay:

```
symcfg show -sid 150 DB-1A -env_data

Symmetrix ID               : 000192600150
Timestamp of Status Data   : 02/03/2009 12:06:42

    Drive Bay

      Bay Name                              :    DB-1A
      Number of Standby of Power Supplies   :    0
      Number of Drive Enclosures            :    8

      Status of Contained Modules
        Standby Power Supplies

        Enclosure Number                    :    1
```

```
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    2
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    3
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    4
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    5
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    6
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    7
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal

Enclosure Number                  :    8
  Enclosure State                 :    Normal
  LCC-A                           :    Normal
  LCC-B                           :    Normal
  PS-A                            :    Normal
  PS-B                            :    Normal
```

Example: Array with system bay with up to eight DAEs

Sample Output for Symmetrix Arrays running Enginuity 5874 containing system bay with up to eight disk array enclosures (DAEs):

Note: In the following example, there is no Drive Bay information.

```
symcfg -sid 139 list -env_data

Symmetrix ID               : 000194900139
Timestamp of Status Data   : 02/26/2009 14:35:57

    System Bay

      Bay Name                              :    SB-1
      Number of Standby Power Supplies      :    6
      Number of Drive Enclosures            :    8
      Number of Enclosure Slots             :    1
      Number of MIBE Enclosures             :    2

      Summary Status of Contained Modules
        All Standby Power Supplies          :    Normal
        All Enclosures                      :    Normal
        All Link Control Cards              :    Normal
        All Power Supplies                  :    Normal
        All Enclosure Slots                 :    Normal
        All Power Supplies                  :    Normal
        All Fans                            :    Normal
        All Management Modules              :    Normal
        All IO Module Carriers              :    Normal
        All Directors                       :    Normal
        All MIBE Enclosures                 :    Normal
        All Power Supplies                  :    Normal

    Drive Bays
```

The following example shows detailed system bay output for Symmetrix arrays running Enginuity 5874 or higher containing system bay with up to eight disk array enclosures (DAEs):

```
symcfg show -sid 139 SB-1 -env_data

Symmetrix ID               : 000194900139
Timestamp of Status Data   : 02/26/2009 14:35:57

    System Bay

      Bay Name                              :    SB-1
      Number of Standby Power Supplies      :    6
      Number of Drive Enclosures            :    8
      Number of Enclosure Slots             :    1
      Number of MIBE Enclosures             :    2

      Status of Contained Modules
        Standby Power Supplies
          SPS-4A                            :    Normal
          SPS-4B                            :    Normal
          SPS-1A                            :    Normal
          SPS-1B                            :    Normal
          SPS-2A                            :    Normal
          SPS-2B                            :    Normal

        Enclosure Number                    :    1
          Enclosure State                   :    Normal
```

```
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    2
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    3
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    4
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    5
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    6
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    7
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Number                 :    8
    Enclosure State                :    Normal
    LCC-A                          :    Normal
    LCC-B                          :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal

  Enclosure Slot Number            :    4
    Enclosure Slot State           :    Normal
    PS-A                           :    Normal
    PS-B                           :    Normal
    FAN-A                          :    Normal
    FAN-B                          :    Normal
    FAN-C                          :    Normal
    FAN-D                          :    Normal
    MM-A                           :    Normal
    MM-B                           :    Normal
    IOMC-A                         :    Normal
    IOMC-B                         :    Normal
```

```
  DIR-7                              :    Normal
  DIR-8                              :    Normal

MIBE Name                            :    MIBE-L-2A
  MIBE State                         :    Normal
  PS-A                               :    Normal
  PS-B                               :    Normal

MIBE Name                            :    MIBE-L-2B
  MIBE State                         :    Normal
  PS-A                               :    Normal
  PS-B                               :    Normal
```

Device pools

The `symcfg` command can be used to look at all pools.

The `symcfg list -pools` command lists information for all the types of pools. You can limit the display to certain type of pools by using the options `-snap`, `-rdfa_dse`, or `-thin`. The `-rdfg` *GrpNum* option can be used with the `-rdfa_dse` option to limit the display to the SRDF/A DSE pools that are related to a specified RDF group. When used, the returned data will include pools that are currently associated with the group or pools that have been disassociated from the group but still have the group's data.

When the `-all` option is specified, both enabled and disabled devices are included in the calculations of Free Tracks and Full %. Otherwise, only enabled devices are included. For example, without the `-all` option, the Free Tracks field would include free tracks from all enabled devices and the Full % field would be based on the Enabled tracks. With the `-all` option specified, The Free Tracks field would include free tracks from both enabled and disabled devices and the Full % would be based on the Total Tracks.

The `[-fba] [-ckd3390] [-ckd3380] [-as400]` options can be used to narrow the returned data set to specific emulation type(s).

The -v option is used to list details of each pool in the returned data set. It is equivalent to using the `symcfg show -pool` command on all desired pools.

```
symcfg  [-sid SymmID] [-offline] [-mb | -gb]
        [-i Interval] [-c Count]

   list [-pool [-snap][-rdfa_dse [-rdfg GrpNum]][-thin]
          [-fba] [-ckd3390] [-ckd3380] [-as400] [-all] [-v]]
```

Where:

`-i Interval -c Count` = the interval and count options can be used to look at the status of the pool(s) continuously for a certain period of time.

`-mb | -gb` = by default, the space consumption of devices and pools is shown in the number of Symmetrix cylinders. For the `symcfg list` and `symcfg show` commands, the output can be shown in megabytes or gigabytes by specifying one of these options. The gigabytes display has one decimal point precision.

`-pool` = lists information for Snap, RDFA/DSE and thin pools in a common output.

`-snap` = lists information for Snap pools only.

`-rdfa_dse` = lists information for RDF/A DSE pools only.

`-rdfg` = can be used with `-rdfa_dse` to limit the display to the RDF/A DSE pools that are related to the specified RDF group. This includes pools that are associated

with the group and pools that have been disassociated from the group, but may still have some data for the group.

-thin = displays information about Virtual Provisioning™ thin device pools. Since AS400 emulation is not supported with Virtual Provisioning, the -as400, -ckd3380, and -ckd3390 options are invalid when listing Virtual Provisioning thin device pools.

-all = includes both enabled and disabled devices in the calculations for Free Tracks and Full %. Otherwise, only enabled devices are included.

-fba | -cdk3390 | -ckd3380 | -as400 = filters the pool display to the specified emulation type.

The display of the pool information for the symcfg list pool command includes the pool name, pool type, device emulation type in the pool, tracks usage of the pool as well as pool state. The pool state indicates whether there are any devices enabled in the pool. If yes, pool state is Enabled, otherwise it is Disabled.

Example all pool output

The following example lists all pools in a Symmetrix array:

```
symcfg -sid 343 list -pool

Symmetrix ID: 000190300343

                        S Y M M E T R I X   P O O L S
-------------------------------------------------------------------------------
              Ty                                                          Sta
 Pool          p Dev   Dev             Total  Enabled     Used     Free Full   t
 Name          e Emul  Config         Tracks   Tracks   Tracks   Tracks  (%)   e
------------ -- ----- ------------ -------- -------- -------- -------- ---- ---
DEFAULT_POOL SN FBA   Mixed         2324370  2324370        0  2324370    0 Ena
DEFAULT_POOL SN 3390  Unknown             0        0        0        0    0 Dis
DEFAULT_POOL SN 3380  Unknown             0        0        0        0    0 Dis
DEFAULT_POOL SN AS400 Unknown             0        0        0        0    0 Dis
xt_snap_2way SN FBA   2-Way Mir      198000   132000        0   132000    0 Ena
range_t_pool SN N/A   Unknown             0        0        0        0    0 Dis
range_pool2  SN FBA   2-Way Mir      198000   198000        0   198000    0 Ena
xt_dse_fba   RS FBA   2-Way Mir      132000    66000        0    66000    0 Ena
xt_tp_2way   TH FBA   2-Way Mir       92160    61440    12080    49360   19 Ena
xt_tp_raid5  TH FBA   RAID-5(3+1)     57480    28740     3380    25360   11 Ena
HR_THIN_R5   TH FBA   RAID-5(7+1)    107520    92160     7880    84280    8 Ena
tp_pool      TH FBA   2-Way Mir      552420   552420        0   552420    0 Ena

Total                              -------- -------- -------- -------- ----
Tracks                            3661950  3455130    23340  3431790    0

Legend for Pool Types:

    SN = Snap,  RS = Rdfa DSE  TH = Thin
```

Example Virtual Provisioning pool output

To list only the Virtual Provisioning thin device pools in this output format, use the `-thin` option. To list thin device pools that have an emulation type of FBA, the CLI option `-fba` should be used.

Note: Beginning with Solutions Enabler V6.5, only FBA emulation is supported. The command specifies the `-gb` option to show the output in gigabytes.

```
symcfg -sid 343 list -pool -thin -fba -gb

Symmetrix ID: 000190300343

                        S Y M M E T R I X   P O O L S
-------------------------------------------------------------------------------
            Ty                                                            Sta
 Pool        p Dev   Dev             Total  Enabled     Used     Free Full   t
 Name        e Emul  Config            GBs      GBs      GBs      GBs  (%)   e
------------ -- ----- ------------ -------- -------- -------- -------- ---- ---
xt_tp_2way   TH FBA   2-Way Mir         1.9      1.9      0.4      1.5   19 Ena
xt_tp_raid5  TH FBA   RAID-5(3+1)       0.9      0.9      0.1      0.8   11 Ena
HR_THIN_R5   TH FBA   RAID-5(7+1)       2.8      2.8      0.2      2.6    8 Ena
tp_pool      TH FBA   2-Way Mir        16.9     16.9      0.0     16.9    0 Ena

Total                               -------- -------- -------- -------- ----
GBs                                     22.4     22.4      0.7     21.7    3

Legend for Pool Types:

    SN = Snap,  RS = Rdfa DSE TH = Thin
```

Example Virtual Provisioning detailed pool output

For Virtual Provisioning thin device pools, there is some information that is more relevant to just those pools than to Snap and RDFA/DSE pools. To display this information, specify the `-detail` option as shown in the example below:

```
symcfg list -sid 343 -pool -thin -mb -detail

Symmetrix ID: 000190300343

                   S Y M M E T R I X   T H I N   P O O L S
-------------------------------------------------------------------------
 Pool        Dev   Dev             Total  Enabled    Alloc Alloc  Subs
 Name        Emul  Config            MBs      MBs      MBs   (%)   (%)
------------ ----- ------------ -------- -------- -------- ----- -----
xt_tp_2way   FBA   2-Way Mir        1920     1920      377    19  3596
xt_tp_raid5  FBA   RAID-5(3+1)      1796      898      105    11     0
HR_THIN_R5   FBA   RAID-5(7+1)      3360     2880      246     8    88
tp_pool      FBA   2-Way Mir       17263    17263        0     0     0

Total                           -------- -------- -------- ----- -----
MBs                                24339    22961      729     3   312
```

The Virtual Provisioning pool output has the following additional information in it that is particularly relevant for thin devices and thin device pools:

- **Total MBs**: Indicates the total capacity of the pool. This includes the sum of the enabled and disabled data devices in the pool.

- **Enabled MBs**: Shows the total capacity of the enabled data devices in the pool. For the pool HR_THIN_R5, the enabled capacity is less than the total capacity. This indicates that there are some data devices in this pool, that are in the disabled state.
- **Alloc MBs**: Shows the amount of space that has been allocated to thin devices bound to this pool. In the example, the HR_THIN_R5 pool has enabled data devices that total 3360 MBs. Of this, 246 MBs have been allocated to accommodate the writes happening to the thin devices that are bound to this pool.
- **Alloc %**: Shows the percentage of space that has been allocated in the pool.
- **Subs %**: Stands for *subscribed percentage*. This indicates the ratio of the capacity of the thin devices that are bound to this pool with respect to the total enabled capacity of this pool. The ratio is expressed as a percentage. The figure 88 in the HR_THIN_R5 column indicates that the capacity of the thin devices bound to this pool is 88% of the capacity of the enabled data devices in the pool. The figure of 3596 in the pool xt_tp_2way indicates that even though the enabled capacity of this pool is 1920 MBs, the total capacity of the thin devices bound to this pool is 3596% of this number. In other words, approximately 67 GB of thin devices are bound to this pool.

DATA devices and SAVE devices

The `symcfg` command can be used to return details about all DATA devices, SAVE devices, and their pool membership. The `-RANGE` and `-nonpooled` options can be used to filter the resulting output.

```
symcfg [-sid SymmID] [-offline] [-mb | -gb]

list [-savedevs [-fba] [-ckd3390] [-ckd3380] [-as400]
          [-nonpooled] [-RANGE SymDevStart:SymDevEnd]]

        list [-datadev [-fba] [-nonpooled]
          [-RANGE SymDevStart:SymDevEnd]]
```

When the `symcfg list` command is used to return pool information, the returned data will include pools that are currently associated with the group, or pools that have been disassociated from the group but still have the group's data.

When the `-all` option is specified, both enabled and disabled devices are included in the calculations of Free Tracks and Full %. Otherwise, only enabled devices are included. For example, without the `-all` option, the Free Tracks field would include free tracks from all enabled devices and the Full % field would be based on the Enabled tracks. With the `-all` option specified, the Free Tracks field would include free tracks from both enabled and disabled devices, and the Full % would be based on the Total Tracks.

DATA device example

The following example returns a list of DATA devices for a specific Symmetrix array:

```
symcfg -sid 343 list -datadev -mb

Symmetrix ID: 000190300343

                S Y M M E T R I X   D A T A   D E V I C E S
-------------------------------------------------------------------------------
     Dev   Dev           Pool                              Total      Used Full
Sym  Emul  Config        Type Pool Name    State            MBs       MBs  (%)
---- ----- ------------ ---- ------------ ----- --------- --------- ----
01C9 FBA   2-Way Mir     TH  tp_pool       Ena      17263         0    0
01CA FBA   2-Way Mir      -  -             Dis      17263         0    0
01CB FBA   2-Way Mir      -  -             Dis      17263         0    0
01CC FBA   2-Way Mir      -  -             Dis      17263         0    0
01FB FBA   2-Way Mir      -  -             Dis        468         0    0
01FC FBA   2-Way Mir      -  -             Dis        468         0    0
022A FBA   RAID-5(3+1)   TH  xt_tp_raid5   Ena        449        48   10
022B FBA   RAID-5(3+1)   TH  xt_tp_raid5   Ena        449        57   12
022C FBA   RAID-5(3+1)   TH  xt_tp_raid5   Dis        449         0    0
022D FBA   RAID-5(3+1)   TH  xt_tp_raid5   Dis        449         0    0
022E FBA   RAID-5(7+1)   TH  HR_THIN_R5     Dis        480         0    0
022F FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480         0    0
0230 FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480        46    9
0231 FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480        51   10
0232 FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480        46    9
0233 FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480        45    9
0234 FBA   RAID-5(7+1)   TH  HR_THIN_R5     Ena        480        57   11
0235 FBA   2-Way Mir      -  -             Dis       1920         0    0
0236 FBA   2-Way Mir     TH  xt_tp_2way    Ena       1920       377   19
0245 FBA   2-Way Mir     TH  xt_tp_2way    Dis        480         0    0
0246 FBA   2-Way Mir     TH  xt_tp_2way    Dis        480         0    0
0247 FBA   2-Way Mir      -  -             Dis        480         0    0
024A FBA   RAID-5(3+1)    -  -             Dis        449         0    0
024B FBA   RAID-5(3+1)    -  -             Dis        449         0    0
024C FBA   RAID-5(3+1)    -  -             Dis        449         0    0
024D FBA   RAID-5(3+1)    -  -             Dis        449         0    0

Total                                                --------- --------- ----
  MBs                                                   82222       727    0

Legend for Pool Types:

    SN = Snap,  RS = Rdfa DSE,  TH = Thin,   - = N/A
```

The displayed output can be filtered further using the `-nonpooled` option to show only the DATA or SAVE devices that are not associated with a pool, and the -RANGE option to display only a range of devices. The following example shows how the `-RANGE` option can filter the resulting output:

```
symcfg list -savedev -sid 343 -gb -RANGE 0147:0150

Symmetrix ID: 000190300343

          S Y M M E T R I X   S N A P   S A V E   D E V I C E S
-------------------------------------------------------------------------------
     Dev   Dev           Pool                              Total      Used Full
Sym  Emul  Config        Type Pool Name    State            GBs       GBs  (%)
---- ----- ------------ ---- ------------ ----- --------- --------- ----
0147 FBA   2-Way Mir     SN  xt_snap_2way  Dis        2.0       0.0    0
0148 FBA   2-Way Mir     SN  xt_snap_2way  Ena        2.0       0.0    0
0149 FBA   2-Way Mir     SN  xt_snap_2way  Ena        2.0       0.0    0
014A FBA   2-Way Mir     SN  DEFAULT_POOL  Dis        2.0       0.0    0
014B FBA   2-Way Mir     SN  DEFAULT_POOL  Dis        2.0       0.0    0
014C FBA   2-Way Mir     SN  DEFAULT_POOL  Dis        2.0       0.0    0
```

```
014D FBA   2-Way Mir     SN  range_pool2   Ena         2.0        0.0    0
014E FBA   2-Way Mir     SN  range_pool2   Ena         2.0        0.0    0
014F FBA   2-Way Mir     SN  range_pool2   Ena         2.0        0.0    0
0150 FBA   2-Way Mir     SN  DEFAULT_POOL  Ena         2.0        0.0    0

Total                                                ---------  --------- ----
  GBs                                                     20.1        0.0    0


Legend for Pool Types:

    SN = Snap,  RS = Rdfa DSE,  TH = Thin,   - = N/A
```

Thin device information

Using the `-tdev` option in `symcfg list` command returns a list of the thin devices in the system. The displayed list can be filtered by emulation type or by a pool name to which the thin devices belong. The `-bound` or the `-unbound` option allows the listing of the thin devices to be filtered further. The -unbound option shows all the thin devices that are unbound, while the `-bound` option shows the bound thin devices.

```
symcfg [-sid SymmID] [-offline] [-mb | -gb]
       list [-tdev [-pool PoolName | [-fba] [-bound | -unbound]]
              [-RANGE SymDevStart:SymDevEnd]]
```

Thin device example

The following is an example output of the thin devices within a given range. The output is shown in MBs:

```
symcfg list -sid 343 -tdev -mb -RANGE 01c0:0240

Symmetrix ID: 000190300343

Enabled Capacity (MBs) :      22961
Bound   Capacity (MBs) :      71583

                S Y M M E T R I X   T H I N   D E V I C E S
--------------------------------------------------------------------------------
                                      Pool              Pool              Pool
                    Dev         Total Subs       Alloc Alloc       Wrtn   Wrtn
Sym  Pool Name      Emul          MBs   (%)         MBs   (%)         MBs    (%)
---- ------------ ----- ---------- ----- ---------- ----- ---------- -----
01C0 xt_tp_2way   FBA        17263   899        133     7          0     0
01C1 xt_tp_2way   FBA        17263   899        134     7          0     0
01C2 xt_tp_2way   FBA        17263   899        138     7         28     2
01C3 xt_tp_2way   FBA        17263   899        137     7          0     0
01C4 -            FBA        17263     0          0     0          0     0
01C8 -            FBA        17263     0          0     0          0     0
01F6 -            FBA          937     0          0     0          0     0
01F7 -            FBA         1875     0          0     0          0     0
01FD -            FBA           93     0          0     0          0     0
021E -            FBA          375     0          0     0          0     0
0222 -            FBA          375     0          0     0          0     0
0226 -            FBA          375     0          0     0          0     0
0237 HR_THIN_R5   FBA          187     7          0     0          0     0
0238 HR_THIN_R5   FBA          187     7          1     0          0     0
0239 -            FBA         3072     0          0     0          0     0
023A -            FBA         1171     0          0     0          0     0
023B HR_THIN_R5   FBA         1078    37        192     7          0     0
0240 HR_THIN_R5   FBA         1078    37          1     0          1     0
```

```
Total                    ---------- ----- ---------- ----- ---------- -----
  MBs                       169174   737        740     3         29     1

symcfg -sid 343 list -tdev -pool HR_THIN_R5

Symmetrix ID: 000190300343

Enabled Capacity (Tracks) :      92160
Bound   Capacity (Tracks) :      81000

                S Y M M E T R I X   T H I N   D E V I C E S
--------------------------------------------------------------------------------
                                    Pool            Pool            Pool
                  Dev        Total  Subs      Alloc Alloc       Wrtn  Wrtn
Sym  Pool Name    Emul      Tracks   (%)      Tracks   (%)      Tracks   (%)
---- ------------ ----- ---------- ----- ---------- ----- ---------- -----
0237 HR_THIN_R5   FBA         6000     7         20     0          0     0
0238 HR_THIN_R5   FBA         6000     7         40     0          0     0
023B HR_THIN_R5   FBA        34500    37       6160     7          0     0
0240 HR_THIN_R5   FBA        34500    37         40     0         28     1

Total                    ---------- ----- ---------- ----- ---------- -----
Tracks                      81000    88       6260     7         28     1
```

Allocated Tracks — the number of tracks that have been allocated (and hence these many tracks have been used from the data pool to which this thin device belongs).

Written Tracks — the number of tracks on which data has actually been written. When a user attempts to write data to a thin device, tracks are allocated in chunks. However, all the allocated tracks may not have data written on them. This column indicates the number of tracks on which data has been written.

Showing pool details

You can show detailed information about a specified device pool using the following syntax:

```
symcfg [-sid SymmID] [-offline]

        show -pool <PoolName> <-snap | -rdfa_dse | -thin> [-all]
          [-fba | -ckd3390 | -ckd3380 | -as400]
```

When showing device pool information, you must specify the type of pool by specifying `-snap` (for Snap SAVE device pools), `-rdfa_dse` (for SRDF/A DSE SAVE device pools), or `-thin` (for Virtual Provisioning thin device pools).

By default, all enabled devices are displayed. If the `-all` option is specified, all devices in the pool will be displayed no matter what their status are.

The `[-fba|-ckd3390|-ckd3380|-as400]` options can be used to specify the emulation type of a pool to return. This is needed when the DEFAULT_POOL for Snap operations is specified. If you specify another pool with one of the emulation type options, the emulation type of the pool and the option must match.

If the specified pool is an SRDF/A DSE SAVE device pool, the display will include a section entitled Related RDF group(s) to list the following information:

- RDF group number.
- Pool Rdfg Attribute. Possible values are:
  - Associated — Pool is currently associated with the RDF group.

- Disassociated — Pool has been disassociated with the RDF group, but there may be data from the group spilled over in the pool. For example, the pool may be in the draining state. When the draining is guaranteed complete, the pool is not related to the RDF group anymore and the entry for the RDF group will be removed.

The following example shows the details of an SRDF/A DSE SAVE device pool:

```
symcfg show -pool yan_pool -sid 166 -rdfa_dse

Symmetrix ID                  : 000287400166
Pool Name                     : yan_pool
Pool Type                     : Rdfa DSE
Dev Emulation                 : FBA
Pool State                    : Enabled
# of Devices in Pool          :   3
# of Enabled Devices in Pool  :   2
# of Related RDF Groups       :   3

Enabled SAVE Devices(2):
  {
   ------------------------------------------------------------------
   Sym       Total     Used      Free    Full  Device    Session
   Dev       Tracks    Tracks    Tracks   (%)  State     Status
   ------------------------------------------------------------------
   018B       66000    33000     33000    50  Enabled   Active
   018C       66000    33000     33000    50  Enabled   Active
            --------  --------  --------  ----
   Tracks    132000    66000     66000     0
   MB(s)       4125    2062.5    2062.5
  }

Related RDF Groups(3) :
  {
   -------- ------------
            Pool Rdfg
    RA-Grp  Attribute
   -------- ------------
 2 ( 1)  Associated
 3 ( 2)  Disassociated
27 (1A)  Associated
  }
```

Thin device data pool details

The `-detail` option can be used with `symcfg show` command to return specific thin device data pool details. With this option, thin devices that are bound to this pool are also displayed. If the `-all` option is added, the disabled devices in the pool are also shown.

```
symcfg -sid 343 show -pool HR_THIN_R5 -thin -mb -detail

Symmetrix ID: 000190300343

Symmetrix ID                  : 000190300343
Pool Name                     : HR_THIN_R5
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : RAID-5(7+1)
Pool State                    : Enabled
# of Devices in Pool          :   7
# of Enabled Devices in Pool  :   6
Oversubscription Limit (%)    :   0

Enabled Devices(6):
  {
```

```
   ---------------------------------------------------
   Sym          Total      Alloc       Free Full  Device
   Dev            MBs        MBs        MBs  (%)  State
   ---------------------------------------------------
   022F           480          0        480    0  Enabled
   0230           480         46        433    9  Enabled
   0231           480         51        428   10  Enabled
   0232           480         46        433    9  Enabled
   0233           480         45        435    9  Enabled
   0234           480         57        422   11  Enabled
          ---------- ---------- ---------- ----
   MBs           2880        246       2633    8
  }

Thin Devices(4):
  {
   ------------------------------------------------------------
   Sym          Total  Subs      Alloc Alloc       Wrtn  Wrtn
   Dev            MBs   (%)        MBs   (%)        MBs   (%)
   ------------------------------------------------------------
   0237           187     7          0     0          0     0
   0238           187     7          1     0          0     0
   023B          1078    37        192     7          0     0
   0240          1078    37          1     0          1     1
          ---------- ----- ---------- ----- ---------- -----
   MBs           2531    88        195     7          1     1
  }
```

RDF group information

To display the Transmit Idle time using `symcfg`, the `-rdfa` option can be specified:

```
symcfg list -rdfg all -rdfa

Symmetrix ID : 000190300016


   S Y M M E T R I X   R D F A   G R O U P S

------- ----------- ------ -----  ---  ---------
RA-Grp  Group       Flags  Cycle  Pri  Transmit
        Name        CSRM T time        Idle Time
------- ----------- ------ -----  ---  ---------
 2 ( 1) RDFDVGROUP  XI-- X    30   33  000:06:45
 3 ( 2) RDFDVGROUP  XI-- X    30   33  000:00:00

Legend:

   RDFA Flags        :
      (C)onsistency   : X = Enabled, . = Disabled, - = N/A
      (S)tatus        : A = Active, I = Inactive, - = N/A
      (R)DFA Mode     : S = Single-session, M = MSC, - = N/A
      (M)sc Cleanup   : C = MSC Cleanup required, - = N/A
      (T)ransmit Idle : X = Enabled, . = Disabled, - = N/A
```

Symmetrix devices

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

A Symmetrix device can map to a part of a physical disk or to an entire disk. The part of a physical disk to which a Symmetrix device is mapped is called a *hypervolume* or a *hyper*. A Symmetrix device may map to multiple hypers (containing identical copies of data) depending on its mirror configuration.

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

Device emulation

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

- Each device has **N** cylinders. The number is configurable (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.)

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

Device type overview

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

Table 4 Device types (page 1 of 2)

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 *gatekeeper* device. The gatekeeper allows you to retrieve configuration and status information from the Symmetrix array without interfering with normal device I/O operations. By default, one of the available Symmetrix devices is designated as a gatekeeper. Alternatively, you can define specific devices to be used as gatekeepers. For more information, refer to Chapter 10, "Gatekeeper Devices."
Metadevices	Allows individual devices to be concatenated to create larger devices. A metadevice consists of a metahead and its member devices. The *metahead* is the first device in the metadevice sequence and is responsible for receiving all incoming commands. It also identifies the entire metadevice. When an incoming command for the metahead is processed, the Symmetrix determines which metadevice member should execute the command. Metahead devices can be added to a device group while a *metamember* 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 *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.
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 *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.
Dynamic RDF devices	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 *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*.
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 *virtual devices* 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 *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide*.

Table 4 Device types (page 2 of 2)

Device type	Description
SAVE devices	Special devices (not mapped to the host) that provide physical storage space for pre-update images or changed tracks during a virtual copy session of TimeFinder/Snap and SRDF/A DSE 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 *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide* and the *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide.*
Device Masking (VCM) devices	Symmetrix devices that have been masked for visibility only 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 *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide.*
DRV devices	Dynamic Reallocation Volume. 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 *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide.*

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 not.

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
--------------------------- ------------ ---------------------------------------
                                                                            Cap
Physical                Sym  SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------ ---------------------------------------

/dev/rdsk/c1t0d0s2      0000 02A:1 01C:C6  Unprotected   N/Grp'd  VCM WD      47
/dev/rdsk/c1t0d1s2      0040 02A:1 16A:D4  Unprotected   N/Grp'd      RW     188
/dev/rdsk/c1t0d2s2      0041 02A:1 01D:D4  Unprotected   N/Grp'd      RW     188
/dev/rdsk/c1t0d3s2      0042 02A:1 16B:C4  Unprotected   N/Grp'd      RW     188
/dev/rdsk/c1t0d4s2      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 Symmetrix array(s). The corresponding output shows the Symmetrix device names (`Sym`), physical device (`Physical`) names, director information, and device-specific information for all devices on the Symmetrix array.

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

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

```
symdev list -sid 64

Symmetrix ID: 000000006164

        Device Name          Directors                  Device
--------------------------- ------------ ---------------------------------------
                                                                            Cap
Sym  Physical               SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------ ---------------------------------------

0002 /dev/sdjt              06A:0 ???:?   TDEV          N/Grp'd       RW   10000
0003 /dev/sdju              06A:0 ???:?   TDEV          N/Grp'd  (M)  RW   10000
0004 Not Visible            ???:? ???:?   TDEV          N/Grp'd  (m)  RW       -
0005 Not Visible            ???:? ???:?   TDEV          N/Grp'd  (m)  RW       -
0006 /dev/sdjv              06A:0 ???:?   R1+TDEV       N/Grp'd       RW   10000
0007 /dev/sdjw              06A:0 ???:?   R2+TDEV       N/Grp'd       RW   10000
0008 Not Visible            ???:? 16D:D5  2-Way Mir     N/A      (DT) RW   10000
0009 Not Visible            ???:? 16A:D1  2-Way Mir     N/A      (DT) RW   10000
000A Not Visible            ???:? 16A:D3  RAID-5        N/A      (DT) RW   10000
000B Not Visible            ???:? 16A:D5  RAID-5        N/A      (DT) RW   10000
000C Not Visible            ???:? 16B:D1  RAID-6        N/A      (DT) RW   10000
000D Not Visible            ???:? 16B:D3  RAID-6        N/A      (DT) RW   10000
000E Not Visible            ???:? 16B:D5  RAID-6        N/A      (DT) RW   10000
```

Example output verbose option

The verbose option (`-v`) with either `symdev list` or `sympd list` will return details about each device (for a single device's details use the `show` argument). The following is an example of the detailed output returned for a host-visible device. This example shows the output for a locally mirrored R22 device:

```
symdev -sid 343 show 173

Device Physical Name     : /dev/sddz

Device Symmetrix Name    : 0173
Device Serial ID         : N/A
Symmetrix ID             : 000190300343

Attached BCV Device      : N/A

Attached VDEV TGT Device : N/A

Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5773
Device WWN               : 60060480000190300343533030313733
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0002
Cache Partition Name     : DEFAULT_PARTITION
```

```
Device Block Size        : 512

Device Capacity
    {
    Cylinders            :       4400
    Tracks               :      66000
    512-byte Blocks      :    4224000
    MegaBytes            :       2063
    KiloBytes            :    2112000
    }

Device Configuration     : RDF2+Mir         (Non-Exclusive Access)

Device is WORM Enabled   : No
Device is WORM Protected : No

SCSI-3 Persistent Reserve: Disabled

Dynamic Spare Invoked    : No

Dynamic RDF Capability   : None

STAR Mode                : No
STAR Recovery Capability : None
STAR Recovery State      : NA

Device Service State     : Normal

Device Status            : Write Disabled   (WD)
Device SA Status         : Ready            (RW)

Front Director Paths (2):
    {
    ------------------------------------------------------------------------
                            POWERPATH  DIRECTOR   PORT             LUN
                            --------- ----------  ---- -------- ---------
    PdevName                Type      Type Num    Sts  VBUS TID SYMM Host
    ------------------------------------------------------------------------
    /dev/sddz               N/A       FA   16D:0  RW   000  00  01E  01E
    Not Visible             N/A       FA   16C:0  RW   000  00  01E  N/A
    }

Mirror Set Type          : [R1 Remote, Data, R1 Remote, Data]

Mirror Set DA Status     : [RW,RW,NR,RW]

Mirror Set Inv. Tracks   : [0,0,0,0]

Back End Disk Director Information
    {
    Hyper Type                          : R1 Remote
    Hyper Status                        : Ready          (RW)
    Disk [Director, Interface, TID]     : [N/A,N/A,N/A]
    Disk Director Volume Number         : N/A
Hyper Number                        : N/A
Mirror Number                       : 1

    Hyper Type                          : Data
    Hyper Status                        : Ready          (RW)
    Disk [Director, Interface, TID]     : [16B, C, A]
    Disk Director Volume Number         :   36 (0x23)
    Hyper Number                        :    5
    Disk Capacity                       : 286102m
    Disk Group Number                   : 0
    Mirror Number                       : 2
```

```
    Hyper Type                              : R1 Remote
    Hyper Status                            : Not Ready        (NR)
    Disk [Director, Interface, TID]         : [N/A,N/A,N/A]
    Disk Director Volume Number             : N/A
    Hyper Number                            : N/A
    Mirror Number                           : 3


    Hyper Type                              : Data
    Hyper Status                            : Ready            (RW)
    Disk [Director, Interface, TID]         : [16A, C, A]
    Disk Director Volume Number             :   36 (0x23)
    Hyper Number                            :    5
    Disk Capacity                           : 286102m
    Disk Group Number                       : 0
    Mirror Number                           : 4
    }


RDF Information
    {
    Device Symmetrix Name                   : 0173
    RDF Type                                : R2
    RDF (RA) Group Number                   : 11                (0A)

    Remote Device Symmetrix Name            : 0111
    Remote Symmetrix ID                     : 000190300342

    R2 Device Is Larger Than The R1 Device  : False
    Paired with a Diskless Device           : False
    Paired with a Concurrent RDF Device     : False
    Paired with a Cascaded RDF Device       : False

    RDF Pair Configuration                  : Normal
    RDF STAR Mode                           : False

    RDF Mode                                : Adaptive Copy
    RDF Adaptive Copy                       : Enabled: Disk Mode
    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     : Enabled
    Prevent RAs Online Upon Power ON        : Enabled

    Device RDF Status                       : Ready            (RW)

    Device SA Status                        : Ready            (RW)
    Device RA Status                        : Write Disabled   (WD)
    Device Link Status                      : Ready            (RW)

    Device Suspend State                    : Offline
    Device Consistency State                : Disabled
    RDF R2 Not Ready If Invalid             : N/A

    Device RDF State                        : Write Disabled   (WD)
    Remote Device RDF State                 : Ready            (RW)

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

    Number of R1 Invalid Tracks             : 3517
```

```
        Number of R2 Invalid Tracks            : 0

        RDFA Information:
            {
            Session Number                     : 10
            Cycle Number                       : 0
            Number of Devices in the Session   : 4
            Session Status                     : Inactive

            Session Consistency State          : N/A
            Minimum Cycle Time                 : 00:00:30
            Average Cycle Time                 : 00:00:00
            Duration of Last cycle             : 00:00:00
            Session Priority                   : 33

            Tracks not Committed to the R2 Side: 0
            Time that R2 is behind R1          : 00:00:00
            R2 Image Capture Time              : Wed May  7 14:23:01 2008
            R1 Side Percent Cache In Use       :  0
            R2 Side Percent Cache In Use       :  0

            Transmit Idle Time                 : 00:00:00
            R1 Side DSE Used Tracks            :  0
            R2 Side DSE Used Tracks            :  0
            }
    }

    RDF Information
        {
        Device Symmetrix Name                  : 0173
        RDF Type                               : R2
        RDF (RA) Group Number                  : 17              (10)

        Remote Device Symmetrix Name           : 0100
        Remote Symmetrix ID                    : 000190300341

        R2 Device Is Larger Than The R1 Device : False
        Paired with a Diskless Device          : False
        Paired with a Concurrent RDF Device    : False
        Paired with a Cascaded RDF Device      : False

        RDF Pair Configuration                 : Normal
        RDF STAR Mode                          : 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    : Enabled
        Prevent RAs Online Upon Power ON       : Enabled

        Device RDF Status                      : Ready           (RW)

        Device SA Status                       : Ready           (RW)
        Device RA Status                       : Write Disabled  (WD)
        Device Link Status                     : Not Ready       (NR)

        Device Suspend State                   : Offline
        Device Consistency State               : Disabled
        RDF R2 Not Ready If Invalid            : Disabled
```

```
Device RDF State                        : Write Disabled  (WD)
Remote Device RDF State                 : Ready           (RW)

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

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

RDFA Information:
    {
    Session Number                      : 16
    Cycle Number                        : 0
    Number of Devices in the Session    : 4
    Session Status                      : Inactive

    Session Consistency State           : N/A
    Minimum Cycle Time                  : 00:00:30
    Average Cycle Time                  : 00:00:00
    Duration of Last cycle              : 00:00:00
    Session Priority                    : 33

    Tracks not Committed to the R2 Side: 0
    Time that R2 is behind R1           : 00:00:00
    R2 Image Capture Time               : Thu May  8 10:25:00 2008
    R1 Side Percent Cache In Use        :  0
    R2 Side Percent Cache In Use        :  0

    Transmit Idle Time                  : 00:00:00
    R1 Side DSE Used Tracks             :  0
    R2 Side DSE Used Tracks             :  0
    }
}
```

Disk geometry details

When the `-geometry` option is specified, the **Effective Device Geometry** field can have three possible values:

- **User Defined**: Indicates the user has defined geometry for this device.
- **Native**: Indicates the current device geometry is the same as the native geometry for the Symmetrix array.
- **Array wide emulation**: Indicates that the array-wide flag for FBA Geometry Emulation is set to Enabled and the effective geometry shown in the output is derived from this setting.

Note: Even if the array-wide setting is turned on, the user can still define the geometry at the individual device level and this setting will take precedence over the array-wide setting. In addition, for non FBA devices, the CLI will indicate N/A with the `-v` and the `-geometry` options.

The following is an example of the device geometry output:

```
symdev -sid 016 show 0016 -geometry -v

    Device Physical Name     : /dev/sdd
    Device Symmetrix Name    : 0016
    Device Serial ID         : 1600016000
    Symmetrix ID             : 000190300016

    Attached BCV Device      : N/A

    Attached VDEV TGT Device : N/A
```

```
Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5771
Device WWN               : 60060480000190300016533030303136
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0001

Device Block Size        : 512

Device Capacity
    {
    Cylinders            :       4400
    Tracks               :      66000
    512-byte Blocks      :    4224000
    MegaBytes            :       2063
    KiloBytes            :    2112000
    }

Effective Device Geometry:    User Defined
    {
    Sectors/Track        :        128
    Tracks/Cylinder      :         15
    Cylinders            :       2000
512-byte Blocks      :    3840000
    MegaBytes            :       1875
    KiloBytes            :    1920000

    }

Device Configuration     : RDF1            (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

STAR Mode                : No
STAR Recovery Capability : Sync_Tgt
STAR Recovery State      : Inactive

Device Service State     : Normal

Device Status            : Not Ready        (NR)
Device SA Status         : Ready            (RW)

Front Director Paths (2):
    …
```

Device service states

The `symdev list` and `symdev list pd` commands provide options that allow the returned data to be limited to a specific service state (`-service_state` option) — `degraded`, `failed`, or `normal`—, or by preceding the service state value with a `not`, such as `-service_state notfailed`. For example, the following command line returns all devices on Symmetrix array 097 that are not in the failed service state:

```
symdev -sid 097 list -service_state notfailed
```

There are three possible device service states:

- Normal
- Failed
- Degraded

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

Device status

There are three possible states to the device status:

- Ready
- Not Ready
- Write Disabled

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

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

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

Device emulation type

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

```
symdev list -inventory

Symmetrix ID: 000187900771

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

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

Filtering device data with 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.

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

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

While the `sympd` command only provides a limited amount of filter options, additional filter options on physical devices can be found by utilizing the `symdev` command with the `list pd` argument and associated options. Refer to "Filtering Symmetrix device data" on page 81 for more information.

Device output options

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

Adding specific device types

Additionally, options can be added to include specific device types and information. To include device masking devices in the output list, specify adding the `-aclx` for arrays running Enginuity 5874 or later, or `-vcm` option for Symmetrix arrays running 5773.x or earlier.

Filtering Symmetrix device data

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

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

```
symdev [-sid <SymmID>] [-offline] [-v] [-resv | -pgr]
            [-wwn] [-all]

          <list | list pd>  [ -FA <#|ALL> [-P <#>] |
            -SA <#|ALL> [-scsi] [-fibre] [-P <#>]]
            [-RANGE <SymDevStart>:<SymDevEnd>]
            [-R1] [-R2] [-R21] [-dldev]
            [-CAP <#>] [-N <#>] [-vcm | -aclx] [-held] [-gige]
            [-ficon] [-escon]
            [-noport|-firstport|-multiport] [-bcv|-nobcv|-drv]
            [-meta] [-hotspare] [-dynamic] [-worm] [-vdev] [-rdfa]
            [-savedev [-nonpooled]] [-raids] [-disk_group <nn>]
            [-rg][-sec_raid] [-unprotected] [-sec_unprotected]
            [-raid1] [-sec_raid1]
            [-raid5 [-protection <3+1 | 7+1>]]
            [-raid6 [-protection <6+2 | 14+2>]]
            [-sec_raid5 [-sec_protection <3+1 | 7+1>]]
            [-sec_raid6 [-sec_protection <6+2 | 14+2>]]
            [-emulation fba|ckd|ckd3390|ckd3380|as400|celerra]
            [-star_mode] [-star_sync_target] [-star_async_target]
            [-half_pair] [-dup_pair] [-bcv_emulation]
            [-reserved | -noreserved] [-cyl] [-geometry_set]
            [-service_state [not]degraded | [not]failed |
                            [not]normal]
            [-tdev [-bound | -unbound]] [-datadev [-nonpooled]]
            [-migr_tgt]
```

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

Using the -all option

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

Filter by director

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

Filter by director port mapping

In addition, you can limit the returned list of Symmetrix devices to one of the following: devices that are mapped to more than one front-end director port (`-multiport`), devices that are not mapped to any front-end director port (`-noport`), or first port information for devices that have more than one port mapping schemas

(`-firstport`). If none of these options are specified, then devices with all director-port relationships are returned.

Filter by device type

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

RAID device information

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

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

```
Device Physical Name     : Not Visible

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

 . . .

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

    RAID-5 Device Information
        {
        Number of Tracks in a Stripe                : 4
        Overall Ready State of RAID-5 Device        : ReadyNoSpare
        Overall WriteProtect State of RAID-5 Device : EnabledNoSpare
        Member Number of the Failing Device         : None
        Mirror Number of the Failing Device         : None
        Mirror Number of the Data Device            : 0
        Member Number that Invoked the Spare        : None
        Mirror Number of the Spare Member           : None
        Disk Director (DA) that Owns the Spare      : None
        Copy Direction                              : N/A
        RAID-5 Hyper Devices (3+1):
            {
            Device : 118B
                {

-------------------------------------------------------------
                Disk     DA       Hyper       Member    Spare       Disk
                DA :IT   Vol#   Num Cap(MB)  Num Status  Status  Grp#
Cap(MB)

-------------------------------------------------------------
                16A:D2    443    72      16    4 RW      N/A        2
139814
                01B:C3    146    72      16    1 RW      N/A        2
139814
```

```
                  01C:C6    296    72      16    3 RW      N/A          0
139814
                  16D:D7    590    71      16    2 RW      N/A          2
139814
                   }
```

Filter by DA, interface, disk, or hypervolume

List host-visible Symmetrix devices that match a specific DA (`-da`), interface (`-interface`), disk (`-disk`), disk group number (`-disk_group`), or hypervolume (`-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/rdsk/c3t1d1s2`, which is Symmetrix device `0046`, enter either:

```
sympd show /dev/rdsk/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 73.

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

Releasing device external locks

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

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

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

If you discover a lock from your host that has been held for well over two 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.

When running Solutions Enabler V7.0 against a Symmetrix array running Enginuity 5874, the `symdev release` command will fail if the user attempts to release locks on devices that are held by the configuration server (such as lock 9). The configuration server can take device locks to run a configuration change session, migration session, for internal use. The error message on the failure will identify the lock owner and, if required, the user can abort the session using the `symconfigure` or `symmigrate` command, whichever is applicable.

Aborting the session, if successful, will release all locks held by the session including device locks. The following examples show three instances of the `symdev release` command when the user attempts to release device locks held by the configuration server.

```
symdev release -sid 343 -range 200:205

Within Symmetrix unit 000190300343, release lock for device 205 (y/[n]) ? y

Device lock is held by a configuration change session <654>. The session should be
aborted to release the lock.

symdev release -sid 343 204

Within Symmetrix unit 000190300343, release lock for device 205 (y/[n]) ? y

Device lock is held by a migration session <654>. The session should be aborted to
release the lock.

symdev release -sid 343 204

Within Symmetrix unit 000190300343, release lock for device 205 (y/[n]) ? y

Device lock is held by an internal config change session. The session should be
aborted to release the lock.
```

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` *`SymDevStart:SymDevEnd`*).

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

- `list` — Provides a list of available disks. You can use various options to expand and restrict your list output. Use the `-v`, `-hypers`, and `-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 (`02A`):

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

Symmetrix ID                : 000187900039
Disks Selected              : 7

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

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

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

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

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

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

  Failed Disk               : False
```

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

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

Disk gaps

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

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

Disk groups

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

Hot spares

Note: Hot spares are not supported on Symmetrix arrays running Enginuity 5874.

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

```
symdisk list -v -spare_info

<...>
  Hot Spare                    : False

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

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

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

  Hot Spare                    : True

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

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

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

5

User Authorization

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

User Authorization operations overview

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

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

User Authorization is configured independently for each Symmetrix array.

Roles

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

There are the Roles defined in Solutions Enabler:

- **None** — Provides no capability at this time.
- **Monitor** — Performs read-only (passive) operations on a Symmetrix array excluding the ability to read the audit log or Access Control definitions.
- **StorageAdmin** — Performs all management (active or control) operations on a Symmetrix array and modifies GNS group definitions in addition to all Monitor operations.
- **Admin** — Performs all operations on a Symmetrix array, including security operations in addition to all Monitor operations.
- **SecurityAdmin** — Performs security operations (`symaudit`, `symacl`, `symauth`) on a Symmetrix array in addition to all Monitor operations.
- **Auditor** — Grants the ability to view, but not modify, security settings for a Symmetrix array (including reading the audit log, `symacl list`, and `symauth`) in addition to all Monitor operations. This is the minimum role required to view the Symmetrix audit log.

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

To list the Roles supported by an array, enter:

```
symauth list -roles

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


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

Users

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

Authentication types

The following authentication types are supported at this time:

- `D` — Indicates a user authenticated by a Windows domain. The qualifier specifies the domain or realm name. For example:

  `D:sales\putman` — User putman logged in through a Windows domain sales.

- `H` — Indicates a user authenticated (by logging in) to some host. On Windows, this would correspond to logging into a local account on the host. The qualifier specifies the hostname. For example:

  `H:jupiter\mason` — User mason logged in on host jupiter.

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

UserName format

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

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

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

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

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

Identify the current username

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

```
symauth show -username

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

How to set up User Authorization

User Authorization user-to-role mappings should be set up prior to enabling. However, at a minimum, you must create at least one mapping for yourself to Admin or SecurityAdmin. You can only enable User Authorization if you (as shown by `symauth show -username`) are already in the user-to-role mappings with a role of either Admin or SecurityAdmin.

The following example provides an overview of the recommended steps for setting up User Authorization:

1. Create the user-to-role mappings. At a minimum, you should map users to the Admin and SecurityAdmin roles when you initially configure User Authorization. These roles provide the ability to perform authorization control operations. You must have one of these user-to-role mappings created before you can successfully enable User Authorization. This can be done using a command file, or, on a UNIX host, by supplying a redirection of STDIN, as shown below:

   ```
   symauth -sid 1234 commit <<!
   assign user H:jupiter\laura to role Monitor;
   assign user D:Eng\neil to role Monitor;
   assign user D:Eng\dave to role Admin;
   assign user D:Eng\steve to role Admin;
   assign user D:Eng\doug to role StorageAdmin;
   assign user D:Eng\bob to role StorageAdmin;
   assign user D:Eng\bill to role SecurityAdmin;
   !
   ```

   In the example above, seven user-to-role mappings have been created, including two Admin roles (D:Eng\dave and D:Eng\steve) and one SecurityAdmin role (`D:Eng\bill`).

2. Once the user-to-role mappings are created, you can then enable User Authorization (assuming you remembered to assign yourself to the Admin or SecurityAdmin role), as shown below:

   ```
   symauth -sid 097 enable
   ```

   In the example above, User Authorization is enabled on Symmetrix array 097. Specifying a Symmetrix ID is optional when enabling User Authorization.

Backing up and Restoring the User Authorization Database

The backup operation saves the contents of the user authorization database from a Symmetrix to the specified file. Use the following command syntax to back up your User Authorization database, where *SymmID* is the 12-character ID that specifies the Symmetrix array and *BackupFile* is the fully qualified path and filename of the backup that will be created:

```
symauth -sid SymmID backup -f BackupFile
```

The restore operation re-initializes the User Authorization database on a Symmetrix array from a previously generated backup file and then re-enables User Authorization. The specified file must be created by an earlier backup operation from the same or a different Symmetrix array. Use the following command syntax to restore a previously created backup file, where *SymmID* is the 12-character ID that specifies the Symmetrix array to restore the file and *BackupFile* is the fully qualified path and filename of an existing backup file:

```
symauth -sid SymmID commit -restore -f BackupFile [-noprompt]
```

The restored file must assign the current user a role of Admin or SecurityAdmin; if it does not, the final restore step, which re-enables User Authorization will fail. If this occurs, assign a role of Admin or SecurityAdmin to the current user and manually enable User Authorization. Alternatively, you can have a user with Admin or SecurityAdmin privileges re-enable User Authorization on the array.

User Authorization management

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

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

Enable and disable User Authorization

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

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

```
symauth -sid 1234 enable
```

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

Set enforcement

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

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

This value can also be set using a command file as described in, "Command file usage" on page 98.

Set server policy

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

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

To change the server policy to trust clients, enter:

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

The `trust_client` option is the default value.

This value can also be set using a command file as described in, "Command file usage" on page 98.

Modify user to Role mappings

Users can be assigned a specific role using the `symauth` command by either supplying the user-to-role mappings via a command file, or, on a UNIX host, by supplying a redirection of STDIN. Each line of the command file must contain a valid username, a supported Role (refer to "Roles" on page 92 for supported Roles), and end with a semicolon. For further details on using the command file, refer to "Command file usage" on page 98.

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

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

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

Commit using STDIN

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

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

Command file usage

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

Adding a user to a Role:

```
assign user <UserName> to role <RoleName>;
```

Reassigning an existing user to Role:

```
reassign user <UserName> to role <RoleName>;
```

Deleting a user:

```
delete user <UserName>;
```

Set enforcement:

```
set enforcement [advise | enforce];
```

Set server policy:

```
set server_policy [trust_client | validate_client];
```

Where:

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

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

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

Preview and Commit actions

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

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

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

User Authorization monitoring

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

View User Authorization policies

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

```
symauth -sid 1234 list

Symmetrix ID: 000000001234

Authorization Control   : Enabled

Time Enabled            : Thu Jun   8  15:02:14 2006
Time Disabled           : Thu Jun   8  12:01:44 2006
Time Updated            : Thu Jun   8  15:02:14 2006

Authorization Mode      : Advise
Server Policy           : Trust clients
```

View user-to-role mappings

List the user-to-role mappings currently established.

```
symauth -sid 1234 list -users

Symmetrix ID: 000000001234

  Role name             Username
  -----------------     ----------------------------------------
  Admin                 moe
  Admin                 D:Eng\moe

  StorageAdmin          larry

  Monitor               curly
```

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

View Roles supported

To list the Roles supported by an array, enter:

```
symauth list -roles

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


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

4

Host-based Access Control

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

Symmetrix Access Control operations overview

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

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

Note: For information on user-based authorization, refer to Chapter 5, "User Authorization."

Access Control command

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

For more information about the syntax of the `symacl` command, refer to the *Solutions Enabler SYMCLI Command Reference HTML Help*.

Access Control database

A Symmetrix-based access control database (shown in Figure 2) contains all the mechanisms or information to govern access to Symmetrix access pools.

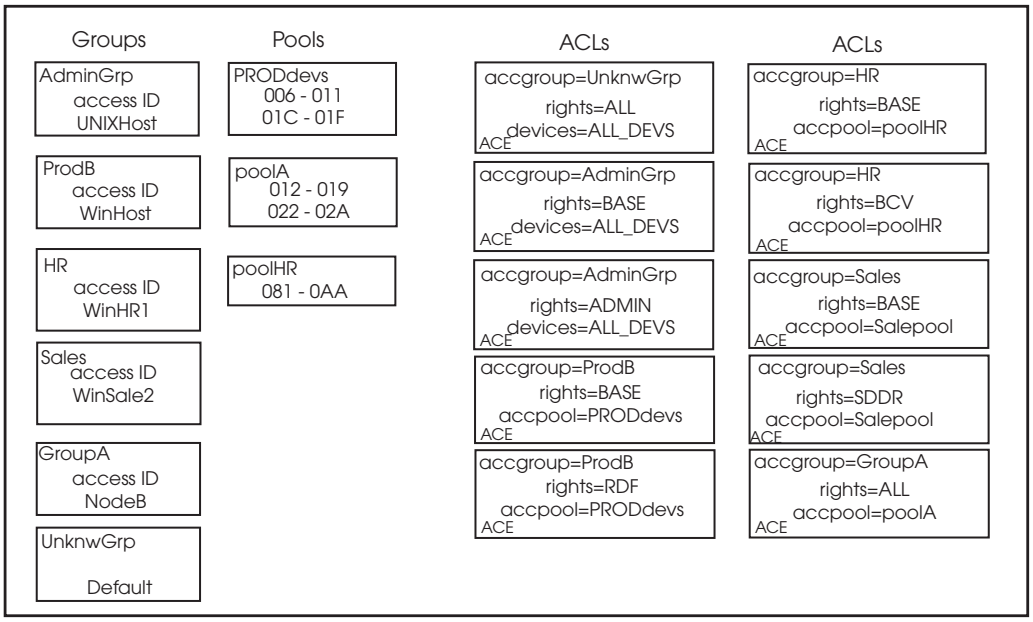


Figure 2 **ACLs and ACEs in the Symmetrix Access Control database**

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

- **Access Control groups** — Unique *access IDs* and *names* are assigned (together) to hosts and then sorted into access control *groups* according to similar needs (determined by an Administrator). Access groups are allowed to act on access pools based on *permissions* (*access types*) granted by the Administrator. The unique host ID for open systems can be viewed by running `symacl -unique`.
- **Access pools** — Permissions (or access types), such as BCV, SRDF, ADMIN, are assigned to allow a host to perform certain Solutions Enabler functionality on a specified set of devices. These sets of devices are referred to as access pools or `accpool`.
- **Access Control Entries (ACEs)** — Once the group and access pool mechanisms are established, the *access control entries* (ACEs) are created, which grant permissions to these pools. The ACEs of the various access groups (along with groups and pools) are managed and stored in the access control database.
- **Access Control Lists (ACLs)** — A group of ACEs that are associated with the same group. For example, in Figure 2, the ACEs that are associated to group HR comprise an ACL in the database.

Figure 3 on page 104 illustrates a Symmetrix site that is managing the permissions of various types of host connections to the Symmetrix devices. Each Symmetrix array has its own access control database, which are separately managed by `symacl`.

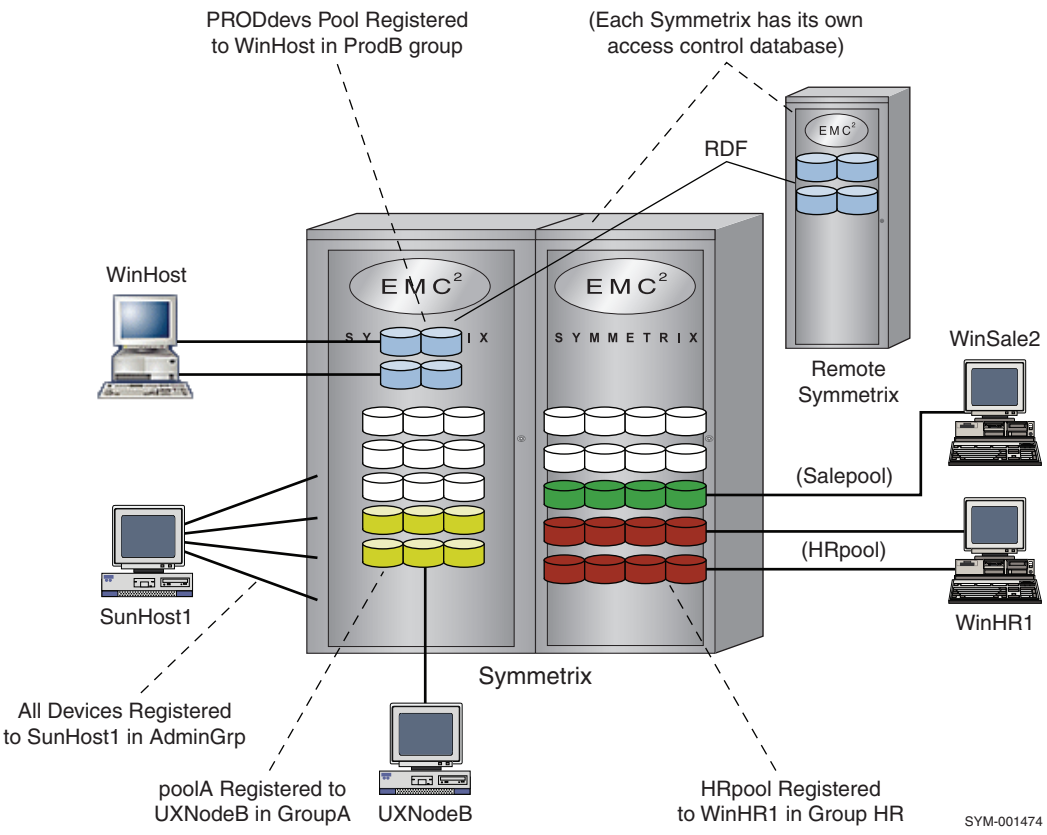


Figure 3 **A Symmetrix site using Access Control**

Alternate Access IDs for x86 platforms

On x86 based platforms, some security information and information used to construct a host's Access ID is stored on disk in a pair of lockbox files (primary and backup). These files are encrypted and checksumed. If the primary copy is corrupt, the backup is used instead.

Note: Do not delete these files.

These files are stored as follows:

Table 5 **Lockbox file location**

Platform	Path
Unix	`/var/symapi/config/lockboxp` `/var/symapi/config/lockboxb`
Win32	`c:\Program Files\EMC\SYMAPI\config\options\lockboxp` `c:\Program Files\EMC\SYMAPI\config\options\lockboxb`

Recommended backup

Since these files are critical to maintaining Access ID information, it is recommended that you maintain a secure backup of these files so that your Access IDs could be restored if these files, or the disks they reside on should be lost or damaged.

How to create or modify Access Control data

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

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

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

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

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

Command file execution

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

- Preview
- Prepare
- Commit

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

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

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

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

Minimum Access Control configuration

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

Table 6 Initial access groups set

Access groups	ID names
AdminGrp	SunHost1 ACCPIN
UnknwGrp	unknown

The name `SunHost1` shown in the Table 6 (associated with `AdminGrp`) is the ID name that designates the machine from which you administer access controls. To perform prepare, commit, and release actions, an access ID (PIN) with the name `ACCPIN` was created in the delivered setup. This ID name is also used for the value setting of SYMCLI_ACCESS_PIN. For more information, see "How to create and manage access groups" on page 107.

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

Initial ACL setup

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

Table 7 Initial delivered ACLs setup

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

Note: The access type ALL excludes ADMIN privileges.

VLOGIX behavior

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

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

How to create and manage access groups

Typically various sets of users tend to use the same applications that utilize common Solutions Enabler features from a given host. They typically require the same device resources and permissions of access to these shared devices. For this reason, hosts are registered in groups identified with a group name, which serves as a root for all ACEs in the group (see Figure 2 on page 103).

Access groups contain groups of access IDs and their ID names. Any ID and name must belong to just one group and are entered into the database together. For ease of management, it is highly recommended that you choose an access ID name that best associates with the particular host in use. For example, SunHost1 is more appropriate than a name such as JRSMITH.

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

Setting up an access group

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

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

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

Create a group

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

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

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

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

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

Add host access IDs to a group

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

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

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

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

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

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

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

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

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

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

User access IDs (PINs) for the AdminGrp

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

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

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

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

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

Note: User access IDs may only be set for the `AdminGrp` access group. Setting an ID for other groups will not return an error or prompt for a PIN in most scenarios.

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

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

Editing and managing an access group

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

Remove an access ID from a group

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

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

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

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

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

Move an access ID to another group

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

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

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

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

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

Remove all ACEs from a group

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

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

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

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

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

Delete a group

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

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

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

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

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

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

How to create and manage limited access to access pools

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

Note: Once an access pool is created, any host in the access group UnknwGrp is denied access to `symmask` commands. In addition, when a host in UnknwGrp calls the `symmaskdb list database` command, only its own devices will be returned in the list. To provide hosts in access group UnknwGrp with full access to the `symmask` commands, grant the access type VLOGIX to access group UnknwGrp. For more information on granting access types, refer to "Grant permissions to an access group" on page 113.

Setting up access pools

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

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

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

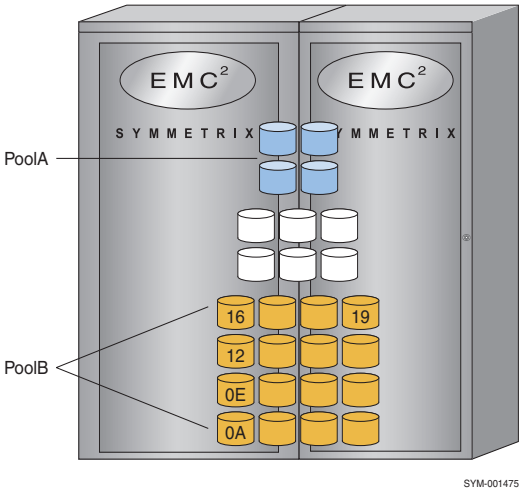


Figure 4 **Creating access pools**

In this Figure 4, 16 devices are assigned to PoolB.

Create a pool

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

```
create accpool PoolB;
```

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

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

Add devices to a pool

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

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

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

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

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

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

Editing and managing access pools

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

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

Remove devices from a pool

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

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

For example, in the command file (**removedevs.cmd**), to assign Symmetrix device names 16 through 19 for removal in PoolB as shown in Figure 5, enter:

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

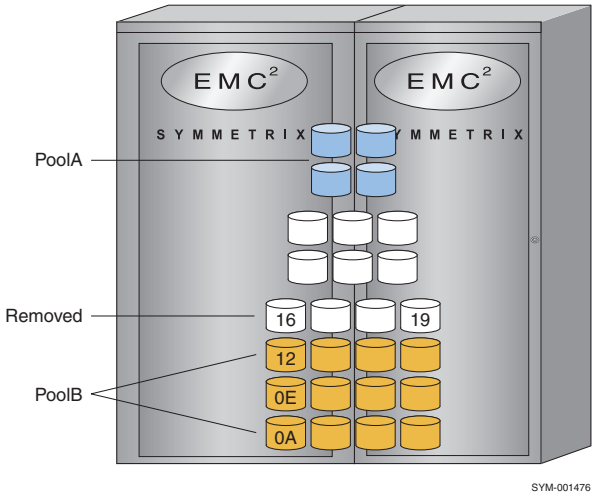


Figure 5 Removing devices from a pool

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

Delete a access pool

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

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

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

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

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

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

How to create and manage access control entries

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

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

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

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

Grant permissions to an access group

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

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

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

AccessTypes: Available Permissions

The possible permissions values are listed and described in Table 8.

Table 8 Permissions: AccessType (page 1 of 3)

Permissions (*AccessType*)	Description	Commands affected
ADMINb	Grants administrator privilege to grant/deny access control entries to hosts and users.	symacl <prepare, commit, release, list, show>
ADMINRDb	Grants read-only access to all access control information.	symacl <list, show>
ALLb	All possible access types granted except ADMIN and ADMINRD. Must be directed to ALL devices.	All
BASE	Allows the discovery of devices and to obtain states and statistics from the Symmetrix array (directors and devices).	Base component commands symevent, symcfg, symaudit, symld, symdev, symcg <list, show>, symdisk, symstat, symipsec (Policy priority list, Policy get, and IPSec statistics get operations symmigrate query and list
BASECTRL	Allows base control operations on devices and device groups.	symld <controls>, symdg <controls>, symcg <controls>
BCV	Allows TimeFinder (BCV) and Clone control and status operations.	symbcv, symmir <controls>, symclone <controls>
CACHCTRL	Allows Cache control operations concerning LRU partition management.	symqos <set LRU >

Table 8 Permissions: AccessType (page 2 of 3)

Permissions (*AccessType*)	Description	Commands affected
CFGDEV	Allows powerful configuration control operations that manage various types of configuration changes on devices in the Symmetrix.	symconfigure Allows the following types of operations: • Convert device configurations (BCV, RDF, DRV) • Convert device configuration (changing mirroring) • Set device attributes • Set device emulation • Metadevice management
CFGSYM b	Allows access to set Symmetrix attributes, set port flags, and swap RA groups with `symconfigure` command. It also affects the `symmaskdb`, `symaccess`, and `symrdf` commands as specified in the next column. Must be directed to ALL devices.	symconfigure, symmaskdb, symrdf, symaccess Allows the following types of operations: • Hot Spare management • SAVE device pool create/delete • SAVE device pool member management • Enable/disable RDFA • Set matrix across Symmetrix array • Add/remove dynamic RDF groups • Change director supporting a dynamic RDF group • Set the link limbo value for a dynamic RDF group • Modify User Authorization settings via `symauth` • Converting thin device (adding RDF) • Setting thin dev attributes • Bind/unbind thin device • Configuring thin metadevices • Allocate/free thin device • Creating thin device pools • Adding devices to thin pools • Enabling/disabling devices in thin pools • Removing devices from thin pools • Deleting thin pools • symipsec policy add, replace/modify, and delete • Initialize the VCMDB • Convert the database type • Restore the database • Manage storage groups • Manage port groups • Manage initiator groups • Manage masking views
CHECKSUM	Allows Symmetrix device Double Checksum operations.	symchksum <controls>
CREATEDVb	Allows the creation and deletion of Symmetrix devices (part of `symconfigure`).	symconfigure • Create new devices • Configure disk space (create/map/mask/meta/attributes) • Delete devices • Create thin devices

Table 8 Permissions: AccessType (page 3 of 3)

Permissions (*AccessType*)	Description	Commands affected
DIRCTRL[b]	Allows you to take directors and their ports offline and online. Must be directed to ALL devices. Also allows you to set CHAP authentication.	symcfg <online/offline> • Online/offline RA directors • Online/offline front-end ports symconfigure • Set port flags and attributes symconnect, symaccess • CHAP authentication
ECC[a, b]	Allows the ECC symmetrix agent to run on the requested host.	Not applicable
OPTMZR	Allows user-configurable attributes that may affect the Optimizer and SymmMigrator behavior.	symoptmz <controls> symmigrate <controls>
POWRPATH[a, b]	Access to PowerPath-directed devices in an RDF consistency group. Must be directed to ALL devices.	Not applicable
QOS	Allows the execution of Quality of Service (QOS) performance control operations to manage copy priorities. Excludes LRU cache control functionality.	symqos <set pace>
RCOPY	Manage Open Replicator sessions.	symrcopy
RDF	Allows SRDF control and set operations.	symrdf <control>
SDDF	Allows the DeltaMark (Change Tracker) functionality that monitors track changes.	symchg
SDR	Allows mapping/unmapping of devices to directors/ports for the Symmetrix Disk Reallocation (SDR) feature, Device Masking, and Autoprovisioning Groups.	symconfigure • Manage CKD aliases • Map/unmap devices • Map/unmap thin device symmask, symmaskdb, symaccess • mapping and unmapping
SNAP	TimeFinder/Snapshot allows the creation and management of virtual copy sessions between a source device and multiple virtual (VDEV) target devices.	symsnap
VLOGIX[a, c]	Enables access to Device Masking devices and Autoprovisioning Groups.	symmask, symmaskdb, symaccess

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

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

c. This access type must be granted to ALL devices in a Symmetrix array.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Remove permissions from an access group

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

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

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

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

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

How to obtain access control information

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

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

Listing access control information

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

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

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

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

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

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

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

```
symacl list -acl
```

Showing access control information

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

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

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

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

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

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

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

Getting your host access ID

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

```
symacl -unique
```

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

```
xxxxxxxx-yyyyyyyy-zzzzzzzz
```

For example:

```
12301558-94200021-00347892
```

How to release pending locked sessions

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

```
symacl release -sid SymmID
```

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

Verifying a locked session

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

```
symacl list -v
```

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

Grant/deny access strategies

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

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

Initially grant all permissions to everybody

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

In Figure 6, when the Symmetrix array was initially configured (delivered), a group named UnknwGrp was created for nonregistered hosts (with no ID).

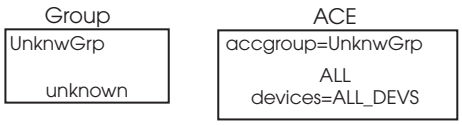


Figure 6 Example default configuration

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

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

Granting and denying default and legacy access

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

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

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

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

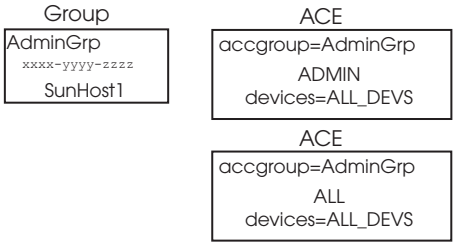
If the default ID is not present, then the access ID of all hosts that need to communicate to the Symmetrix must be registered in the database and granted sufficient permissions to accomplish the functions they need to perform.

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

Establishing an administrator

As shown in Figure 2 on page 103 and Figure 3 on page 104, the delivered setup allows at least one host assigned with administrative (ADMIN) privileges to be designated the administrator of the information in the access control database.

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

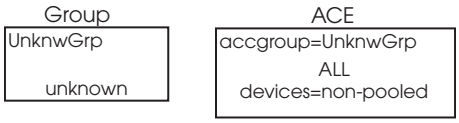


Next, two ACEs were created: one granting ADMIN permissions and one granting ALL permissions to group AdminGrp for all devices in the Symmetrix.

Grant all permissions to the nonpooled devices

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

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



General absolute control strategy

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

Initial setup summary

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

BCV setup strategy

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

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

TimeFinder/Snap setup strategy

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

Source and target strategy

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

SAVE device strategy

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

SRDF setup strategy

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

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

How to back up and restore the access control database

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

Create a backup file

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

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

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

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

Restore the backup file to the access control database

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

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

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

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

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

6 Grouping Devices

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

Overview

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

Grouping types

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

- A *device group* (DG) is a user-defined group comprised of devices that belong to a single Symmetrix array and a single RDF (RA) group. A control operation can be performed on the group as a whole, or on the individual device pairs that comprise it. By default, a device cannot belong to more than one device group and all of the STD devices in a group must reside on the same Symmetrix array. However, if the Symmetrix options file parameter SYMAPI_ALLOW_DEV_IN_MULT_GRPS is enabled, a device can be added to multiple groups. You can use device groups to identify and work with a subset of available Symmetrix devices; obtain configuration, status, and performance statistics on a collection of related devices; or issue control operations that apply to all devices in the specified device group.
- A *composite group* (CG) is a user-defined group comprised of devices that can belong to one or more locally-attached Symmetrix arrays and one or more RDF (RA) groups within a Symmetrix. For more information on composite groups, refer to "Composite groups" on page 140.
  - An *RDF consistency group* is a CG comprised of RDF devices (RDF1, RDF2, or RDF21), 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 Multisession Consistency (MSC), which respects the logical relationships between dependent I/O cycles. Refer to "What is an RDF consistency group?" on page 140 for information on using RDF consistency groups.

Group name services (GNS)

In a default Symmetrix environment, device group and composite group definitions are created through 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 hosts. GNS provides the optional ability to store device and composite group definitions in a shared repository located on each Symmetrix array, which would then automatically become visible to all locally attached hosts. This allows all GNS-enabled hosts to see the same group definitions across your Symmetrix environment, while sharing real-time updates to group definitions and configurations made by other hosts.

For more information on GNS, refer to Chapter 7, "Group Name Services."

Group names and device names

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

When you add a device to a device group, it is given a logical name. This name allows you to refer to the device independently of its physical device name or Symmetrix device name. The name can have up to 31 characters and must be unique within the device group. It is known only within the context of the device group to which the device belongs. You must use this logical name in any SYMCLI command with 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, RDF2, or RDF21). Otherwise, the group type defaults to REGULAR when no type is specified. The following device lists can be maintained in device groups:

- **SRDF device list** — You can group SRDF (Symmetrix Remote Data Facility) standard devices in a device group subject to the following restrictions:
  - You cannot mix SRDF and non-SRDF devices in the same device group.
  - All SRDF devices in a given device group must be either all source (R1), all target (R2), or all concurrent/cascaded (R11, R21, R22) devices.
  - All SRDF devices in a given device group must belong to the same Symmetrix RDF group[1] 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.

- **TimeFinder/BCV device list** — 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. Refer to *EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide* for more information on adding BCV devices to a device group.
- **TimeFinder/Snap virtual device list** — A virtual device is a host-accessible device containing address pointers to the data stored on the source device or a pool of save devices, which indicate where the copy session data is located in the physical storage. You can associate virtual devices (VDEV) paired with standard and BCV devices with any device group. In addition, TimeFinder/Snap control operations can be performed on any virtual device in a device group.

1. RDF groups (also known as RA groups) are configured by the Symmetrix Enginuity software. 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/Clone target list** — Target lists provides a mechanism establishing source (SRC) and target (TGT) devices for TimeFinder/Clone operations. They can be created for both device groups and composite groups. A Target list can contain various types of devices, including STDs or BCV devices (based on a set of rules discussed in "Clone target restrictions" on page 127) and use those devices as targets in Clone operations. Remote Target lists can also be created for remote operations.
- **Gatekeeper device list** — You can *associate* one or more gatekeeper devices with a device group. SYMCLI will use the associated gatekeeper to issue requests to the Symmetrix for control operations on the devices within the specified device group. You can *add* a standard device to a device group. However, the gatekeeper cannot be added to the device group, only associated with a device group. For more information about associating a gatekeeper device with a group, refer to Chapter 10, "Gatekeeper Devices."

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

How to create a device group

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

Creating an empty device group

Device groups are created using the `symdg` command. When you create a device group, you assign it a name and choose its type: REGULAR (non-RDF), RDF1 (RDF source device), RDF2 (RDF target device), or RDF21 (uses 2 mirrors for concurrent RDF operations over two hops). If a type is not specified, the default group created is REGULAR.

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

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

Device restrictions

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

RDF1 and RDF2 type device restrictions

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

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

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

RDF21 type device restrictions

All devices added to an RDF21 device group:

- Must be R21 STD devices. No mixture of STD device types will be allowed.
- Existing rules for adding BCV, VDEV, and TGT devices to RDF21 groups still apply.
- That have the same first RDF group number, must have the same cascaded RDF group number in a cascading RDF configuration.

Virtual device restrictions

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

Clone target restrictions

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

- Non-RDF STDs

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

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

Adding devices to a device group

SYMCLI provides various commands to add standard devices (STD), virtual devices (VDEV or RVDEV), or TimeFinder/Clone target (TGT or RTGT) devices to a device group. Once a device group has been created, SYMCLI commands can be used to add a single device (`symld`), all devices on a Symmetrix not yet assigned to a device group (`symld`), or a list of devices from a file (using `symdg`) to that group. For more information on adding a list of devices from a file, refer to "Device groups: Exporting and importing device lists" on page 134.

Add devices by physical device name

The `symld` command is used to add a single device to a device group. All devices in a device group must be part of the same Symmetrix array. Devices can be added by specifying either the physical device name (`add pd`) or the Symmetrix device name (`add dev`). When adding virtual devices or target devices, specify the Symmetrix device name (*SymDevName*). Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays. The following is the syntax used to add a device:

```
symld -g DgName [-h] [-offline] [-i Interval] [-c Count]

        add pd PdevName [LdevName]

        add dev SymDevName [LdevName][-sid SymmID]
          [<-vdev | -tgt> <-rdf | -hop2>
            [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]]
```

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

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

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

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

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

Adding virtual devices

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

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

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

When adding virtual devices from a Symmetrix array two hops away, specify the `-vdev -hop2 -remote_rdfg` *`RemoteGrpNum`* option along with the Symmetrix device name. This targets the operation to the specified virtual device over RDF links on the remote array two hops away.

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

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

Adding devices to the target list

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

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

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

When adding target devices from a Symmetrix array two hops away, specify the `-tgt -hop2 -remote_rdfg` *`RemoteGrpNum`* option along with the Symmetrix device name. This targets the operation to the specified target device list over RDF links on the remote array two hops away.

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

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

For details on the types of devices that can be added to a device group's target list, refer to "Clone target restrictions" on page 127.

Adding all ungrouped devices

Use `symld` to add all ungrouped devices on a Symmetrix array to a device group. By default, this command will add all standard devices (or local virtual devices) to a device group. However, various optional criteria can be set to specify the types of devices to add to the specified device group. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays. Use the following syntax to specify the types of ungrouped devices to add to a device group:

```
symld -g DgName [-offline] [-i Interval] [-c Count]
   addall [-sid SymmID][-SA <#|ALL>][-P #]
          [<-vdev | -tgt> <-rdf | -hop2>
          [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]]
          [-CAP #] [-N #] [[-range SymDevStart:SymDevEnd]]
          [pd | dev |-host HostName]
```

- All ungrouped 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` *`SymDevStart`*`:`*`SymDevEnd`*`]`), or that are of a specific capacity (`-CAP #`)
- All virtual devices from a local or remote array (`-vdev -rdf -rdfg` *`GrpNum`*)
- All devices are to be added to the target list of the device group for TimeFinder/Clone operations on a remote array (`-tgt -rdf -rdfg` *`GrpNum`*)
- All virtual devices from a remote array two hops away (`-vdev -hop2 -remote_rdfg` *`RemoteGrpNum`*)
- All devices are to be added to the target list of the device group for TimeFinder/Clone operations on an array two hops away (`-tgt -rdf -hop2 -remote_rdfg` *`RemoteGrpNum`*)

Note: In the situation where you have concurrent RDF, whereby there are two Symmetrix arrays on the remote side, you must specify the RDF group number (`-rdfg` *`GrpNum`*) with `-rdf` option. When the `-hop2` option is specified, you must specify a `-remote_rdfg` *`RemoteGrpNum`*.

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 *Solutions Enabler SYMCLI Command Reference HTML Help*.

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

                                                                Number of
Name                Type      Valid  Symmetrix ID  Devs    GKs  BCVs  VDEVs  TGTs

Team1A              RDF1      Yes    000000006164     2      0     0      0     2
Eng1                REGULAR   Yes    000187900039    14      0     5      0     4
Acct_Team23         RDF1      Yes    000187900039     1      0     0      0     1
All                 REGULAR   Yes    000187940256     1      0    16      2     1
```

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

Show details about a device group

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

```
symdg show clonetgtdg
```

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

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

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

Device groups: Exporting and importing device lists

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

Exporting a device list

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

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

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

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

```
symdg export prod2 -f prod2list
```

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

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

Exporting and deleting a device group

The delete option allows you to export the device group membership to a file and delete the existing device group in the same operation. To reinstate the same group again, you can import this list (refer to "Importing device lists" on page 135) 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:

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

Refer to the *EMC Solutions Enabler SYMCLI Command Reference HTML Help* for file format details.

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:

```
symdg export prod2 -f prod2list -delete
```

For information on deleting a device group, refer to "Deleting a device group" on page 139.

Note: The `-rdf` option is not supported when exporting R21 device groups.

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 134).

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

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

Importing OSM lists

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

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

```
symdg file2dg prod3 -f prod_OSMlist
```

Rename, move, copy, and delete device groups

Solutions Enabler also provides the ability to rename, move, and delete existing device groups.

Renaming

Device groups—and the logical device name of devices in a device group—can be renamed.

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
```

Devices

Use `symld` to change the logical name of a device. The name can have up to 31 characters and must be unique within its device group.

For example, to rename the logical name of device `DEV003` to `TEMP3` in device group named `prod`, enter:

```
symld -g prod rename DEV003 TEMP3
```

Moving devices

Use `symld` to move one device, specifying the logical device name, from one existing device group to another existing device group. The source and destination device groups must be compatible types. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays. The following syntax is used to move an individual device:

```
symld -g DgName [-h] [-offline] [-i Interval] [-c Count]

   move LdevName DestDgName [-force] [-rename]
```

For example, to move logical device `DEV003` from device group `prod` to device group `test`, enter:

```
symld -g prod move DEV003 test
```

If there exists a device within the destination device group with the same logical device name as device you wish to move into the destination device group, you must use the `-rename` option to avoid encountering an error. When this option is used, SYMCLI will rename the moved device to the next available logical device name as defined in the SYMCLI_LDEV_NAMING environment variable.

Moving all devices

When choosing to move all devices from one existing device group to another, you can move all devices, or use the additional options to focus the operation to devices that meet a specific criteria. By default, this operation only affects standard devices. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays. However, the other option can be specified to only move specific device types as shown in the following syntax:

```
symld -g DgName [-h] [-offline] [-i Interval] [-c Count]

   moveall DestDgName [-force] [-rename]
          [-SA <#|ALL>][-P #][-CAP #] [-N #]
          [<-vdev | -tgt> [-hop2] | -rvdev | -rtgt]
          [-range SymDevStart:SymDevEnd]
```

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 SymDevStart:SymDevEnd]`) to move. The `-vdev` option adds only the Symmetrix virtual devices to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away. The `-tgt` option adds only the TGTs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.

The source and destination device groups must have compatible types.

For example, to move all virtual devices from device group `prod` to device group `test`, enter:

```
symld -g prod moveall -vdev test
```

Copy devices from DG to DG

Devices from an existing device group can copied into another existing device group of compatible type. When the `copy` action is used, one standard device is copied from the specified source device group to the destination device group. The source and destination device groups must have compatible types. When the `copyall` action is used, copies of all standard devices are copied from the specified source device group to the destination device group. The source and destination device groups must have compatible types. When performing a `-copyall` action, the types or number of devices that are included in the copy can be limited using the various filter options. The following syntax can be used to copy the devices from one device group to another.

```
symld -g DgName [-h] [-offline] [-i Interval] [-c Count]

   copy LdevName DestDgName [-force] [-rename]

   copyall DestDgName [-force] [-rename]
     [-SA <#|ALL>][-P #][-CAP #][-N #]
     [<-vdev | -tgt> [-hop2] | -rvdev | -rtgt]
     [-range SymDevStart:SymDevEnd]
```

Copy devices from a DG to a CG

Devices from an existing device group can be copied (or added) to an existing composite group. Use the following syntax to add devices from an existing device group to an existing composite group:

```
symdg dg2cg DgName CgName
[-bcv [-hop2]|-nobcv|-rbcv|-rrbcv|-brbcv|
-vdev [-hop2]|-rvdev|-tgt [-hop2]|-rtgt]
[-apidb | -rdf_consistency] [-force] [-rename]
```

The following additional options are available:

- `-bcv` adds only the BCVs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-vdev` adds only the Symmetrix virtual devices to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-rbcv` adds only the RBCVs to the composite group.
- `-rrbcv` adds only the RRBCVs to the composite group.
- `-brbcv` adds only the BRBCVs to the composite group.
- `-nobcv` adds only the STDs to the composite group.
- `-rvdev` adds only the Symmetrix remote virtual devices to the composite group.
- `-tgt` adds only the TGTs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-rtgt` adds only the RTGTs to the composite group.

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

Setting consistency

The `[-rvdev | -apidb | -rdf_consistency]` options allow you to choose a consistency method for the CG. If the `-rvdev` parameter is specified, the Symmetrix remote virtual devices will be added. If the `-apidb` parameter is specified, the composite group will only be committed to the SYMAPI configuration database and consistency is not maintained. If `-rdf_conistency` 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 Multisession 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 `rmall`. By default, the `remove` and `rmall` arguments will affect standard devices only. However, the `-vdev` option can be specified to remove just the virtual devices.

For example, to remove logical device `DEV003` from a device group named `prod`, enter:

```
symld -g prod -force remove DEV003
```

In this example, the `-force` option was used to remove the device regardless of its BCV state.

Note: If you remove the only member of a device group, the device group is not automatically deleted. You must execute `symdg delete` to explicitly delete the device group, as described in the "Deleting a device group" on page 139.

Removing all devices

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

```
symld -g DgName [-h] [-offline] [-i Interval] [-c Count]
        rmall [-force] [-SA <#|ALL>] [-P #] [-CAP #] [-N #]
          [-range SymDevStart:SymDevEnd]
          [<-vdev | -tgt> -rdf [-rdfg GrpNum]]
```

Options include specifying devices mapped to a specific front-end (SCSI or Fibre) director number, (`-SA <#|ALL>`), devices mapped to a specific (SCSI or Fibre) director port number (`-P #`), devices of a specified capacity (`-CAP #`), total number of devices (`-N #`), a range of devices (`-range SymDevStart:SymDevEnd`) to remove, or local and remote virtual and target devices (`[<-vdev | -tgt> -rdf [-rdfg GrpNum]]`).

Deleting a device group

Use the following syntax to delete a device group:

```
symdg

   delete DgName [-force]
```

For example, to delete device group named `prod`, enter:

```
symdg delete prod
```

Note: Deleting populated device groups requires the use of the `-force` option.

Composite groups

A Composite Group (CG) is a user-defined group whose STD device members can belong to multiple Symmetrix arrays and RA groups.

A single composite group can contain devices from the following different device lists:

- Standard device list (STD) — Non-BCV devices that are local to the host.
- Local Business Continuance Volume (BCV) list — BCV devices local to the host.
- Local VDEV list (VDEV) — Virtual devices that are local to the host.
- Remote VDEV list (RVDEV) — Virtual devices that are remote.
- Remote BCV list (RBCV) — BCV devices that are to be associated with the remote mirrors of the STD devices.
- BCV-Remote BCV list (BRBCV) — BCV devices that are to be associated with the remote mirrors of the local BCV devices.
- Remote-Remote BCV List (RRBCV) — Remote BCV devices that are to be associated with the remote mirrors of the RBCV devices.
- Local TGT list (TGT) — TimeFinder/Clone target devices that are local to the host.
- Remote TGT list (RTGT) — TimeFinder/Clone target devices that are remote.
- R21 STD devices — R21 devices utilize two mirrors and are considered to be Concurrent RDF devices.

A composite group can be defined as type REGULAR, RDF1, RDF2, or RDF21.

What is an RDF consistency group?

An RDF consistency group is a composite group comprised of RDF devices (RDF1, RDF2, or RDF21 devices) acting in unison to preserve dependent write consistency of a database distributed across multiple SRDF systems. It maintains this consistency by using either Multisession Consistency (MSC) or RDF-ECA.

A group is considered an RDF consistency group if it is a composite group that meets ALL of the following criteria:

- Created as type RDF1, RDF2, or RDF21
- The composite groups must be marked as being registered with the RDF daemon
- Contains STD devices
- Enabled for consistency

Note: For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at http://elabnavigator.emc.com.

Multisession 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 serviced. SRDF/A MSC provides a single coordination point to drive the cycle switch process in all participating Symmetrix arrays.

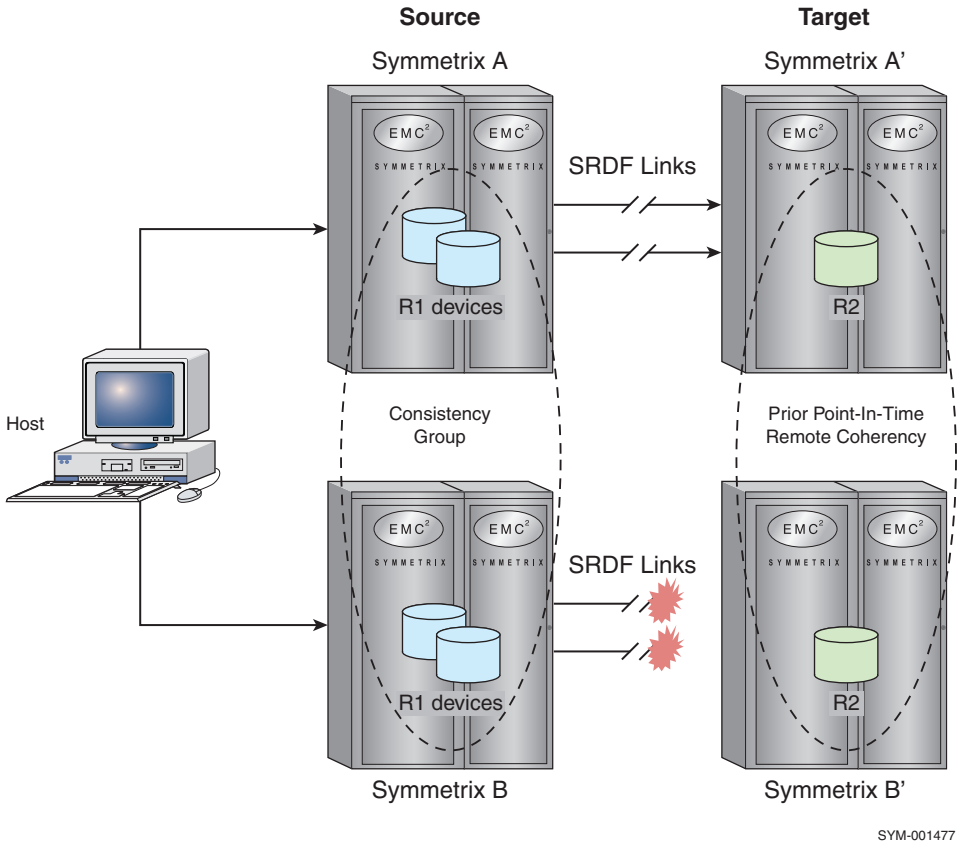


Figure 7 RDF consistency groups

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,r RDF2, or RDF21:

```
symcg [-i Interval] [-c Count]
   create CgName [-type REGULAR | RDF1 | RDF2 | RDF21]
     [-apidb | -rdf_consistency]
```

If you do not specify a type during the `symcg create` operation, the group will be set as a REGULAR group by default. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays.

If `-apidb` is specified, the composite group will not be added to a consistency group (for more information, refer to "Creating composite groups without consistency"). When creating a composite group of type RDF1, RDF2, or RDF21, if the `-rdf_consistency` option is specified, the composite group will be added to the RDF daemon.

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 the `-apidb` parameter is supplied with the `symcg` command.

When a composite group is created without consistency, a flag is set indicating so.

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*. Therefore, you are unable 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.

Moving a composite group to or from MSC

To move a composite group from or to MSC, export the composite group to a file, delete the composite group, and then import the composite group with `-apidb`.

Exporting a composite group to file

Use the following syntax to export a composite group to file:

```
symcg [-i Interval] [-c Count]
   export CgName [-f FileName] [-rdf]
   exportall [-f FileName] [-rdf]
```

Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays.

Note: The `-rdf` option cannot be used on REGULAR or cascaded RDF device groups. Specifying the `-rdf` option will produce an error if any Hop-2 devices are detected in the CG.

Refer to the *EMC Solutions Enabler SYMCLI Command Reference HTML Help* `symcg` command for details on the file format.

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. The resulting file will have as many device description lines as the composite group has members.

Deleting a composite group

The following syntax is used to delete a composite group:

```
symcg [-i Interval] [-c Count]
   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 the composite group is enabled for MSC, you cannot delete it unless the `-force` option is specified. Specifying `-force` does not disable the composite group. You must use `symcg disable` to disable the composite group.

Importing a composite group

The following syntax is used to import a composite group from a previously generated file:

```
symcg [-i Interval] [-c Count]
   import CgName [-f FileName]
     [-apidb | -rdf_consistency] [-rename]

   importall [-f FileName] [-apidb | -rdf_consistency]
```

If the `-apidb` parameter is specified then the composite group will be created in the SYMAPI configuration database only. If the `-rdf_consistency` parameter is specified then the composite group will be created in the SYMAPI configuration database and added to the RDF daemon.

Use the `-rename` option with the `symcg import` command to have the operation result in a newly named composite group.

Adding devices

Once a composite group has been created, standard (STD) devices can be added to it.

Standard devices

The following syntax can be used to add a STD device to a composite group:

```
symcg [-i Interval] [-c Count]
   add pd PdevName [LdevName] -cg CgName
```

or

```
symcg [-i Interval] [-c Count]
   -cg CgName -sid SymmID [-rdf | -hop2]
     [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]

   add dev SymDevName [LdevName] [-vdev | -tgt]
```

When adding devices by specifying a Symmetrix device name, the Symmetrix array's ID (`-sid`) is required and additional options provide greater controls. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the Symmetrix host database and, for RDF control operations, on the local and/or remote Symmetrix arrays.

Composite groups will allow remote BCV devices (RBCVs) on both links to be associated with the RDF group. These RBCVs may be R1 or R2 type BCVs and may also have a remote BCV (RRBCV) associated behind each.

When the `-vdev` option is specified, the devices are added to the virtual device list. For this action to succeed, the device must already be configured as a VDEV. In addition, to add a VDEV to the remote virtual device list, specify the `-rdf` option with the `-vdev` options. If the device is two hops away, specify the `-hop2` option instead.

When the `-tgt` option is specified, the devices are added to the target device list. In addition, to add a target device to the remote target device list, specify the `-rdf` option with the `-tgt` options. If the device is two hops away, specify the `-hop2` option instead.

The RDF group selection option, `-rdfg GrpNum`, can be used on nonconcurrent RDF or concurrent RDF devices within a composite group to limit the scope of the action. A composite group could have two or more nonconcurrent RDF devices on the same

local Symmetrix but in two or more RDF groups. The user could restrict the operation to only those devices in the specified RA groups. Similarly, a composite group could have concurrent RDF devices, which would each, by definition, be in two RDF groups. The user could use RDF group selection to restrict the operation to only one RDF group of the concurrent RDF device.

RDF group selections always refer to the locally attached Symmetrix arrays and RA group, even if they are controlling devices or links that are not on the locally attached Symmetrix. However, when the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa. To specify a device two hops away, specify the `-remote_rdfg` option. This provides the RDF group number to access a two hop device from the first hop.

All devices

The following syntax can be used to add all devices that belong to a specified Symmetrix array and are not already members of a composite group to a specified composite group.

```
symcg [-i Interval] [-c Count]
   -cg CgName [-sid SymmID] [-range SymDevStart:SymDevEnd]
     [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
     [-vdev | -tgt]

   addall [pd | dev [-rdf | -hop2]] [-R1|-R2]
```

If `-vdev` is specified, all VDEVs for the specified Symmetrix will be added to the virtual device list of the composite group. If `-tgt` is specified all devices will be added to the Clone target list of the composite group.

Note: You cannot add devices to a CG enabled for RDF MSC.

Moving or copying devices

Devices from an existing composite group can be moved or copied into another existing composite group of compatible type. When the `move ld` or `moveall` action is used, the devices are removed from the source composite group and added to the destination composite group. When the `copy ld` or `copyall` action is used, copies of the devices will be added to the destination composite group, and the source composite group will remain unchanged. The following syntax can be used to move or copy devices from one composite group to another.

```
symcg -cg CgName [-i Interval] [-c Count]

   move ld LdevName DestCgName [-force] [-rename]
   copy ld LdevName DestCgName [-force] [-rename]

symcg -cg CgName [-sid SymmID] [-range SymDevStart:SymDevEnd]
          [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
          [<-vdev | -tgt> [-hop2] | -rvdev | -rtgt]

   moveall DestCgName [-force] [-symforce]
   copyall DestCgName [-force] [-symforce]
```

Removing standard devices

Devices may be removed from an existing composite group. You can either remove devices individually, remove all devices, or remove all devices meeting the specified criteria.

The following syntax is used to remove a STD device from a composite group:

```
symcg -cg CgName -sid SymmID [-i Interval] [-c Count] [-rdf | -hop2]
          [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   remove dev SymDevName [-force] [-symforce]
```

or

```
symcg remove [-i Interval] [-c Count] [pd PdevName | ld LdevName]
-cg CgName [-force] [-symforce]
```

or

```
symcg [-i Interval] [-c Count]
remove -cg CgName [-force] [-symforce]
```

When removing devices by specifying a Symmetrix device name, the Symmetrix array's ID (`-sid`) is required. RDF group selection option, `-rdfg GrpNum`, can be used on nonconcurrent RDF or concurrent RDF devices within a composite group to limit the scope of the action. A composite group could have two or more nonconcurrent RDF devices on the same local Symmetrix array but in two or more RDF groups. The user could restrict the operation to only those devices in the specified RA groups. Similarly, a composite group could have concurrent RDF devices, which would each, by definition, be in two RDF groups. The user could use RDF group selection to restrict the operation to only one RDF group of the concurrent RDF device. To specify a device two hops away, specify the `-remote_rdfg` option. This provides the RDF group number to access a two-hop device from the first hop.

RDF group selections always refer to the locally attached Symmetrix arrays and RA group, even if they are controlling devices or links that are not on the locally attached Symmetrix. However, when the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa.

If 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.

The following syntax is used to remove all STD devices from a composite group:

```
symcg -cg CgName [-i Interval] [-c Count]
     [-sid SymmID] [-range SymDevStart:SymDevEnd]
     [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
     [-vdev | -tgt]

   rmall [-rdf | -hop2] [-force] [-symforce]
```

If `-vdev` is specified, only all VDEVs will be removed from the VDEV device list. If `-tgt` is specified, only all devices in the target device list will be removed.

If RDF consistency is enabled on a device you wish to remove, the command will fail unless the `-force` flag is used. If RDF consistency cannot be disabled, the command will fail unless the `-symforce` flag is used.

When the `-rdf` option is specified, the operation occurs on the remote array, switching the R1 to the R2 and vice versa.

Renaming a composite group

To rename a composite group, enter:

```
symcg [-i Interval] [-c Count]
   rename OldCGName NewCGName
```

Note: You cannot rename an MSC RDF consistency CG that has been enabled.

Setting RDF group attributes

Set a logical name to be associated with RDF groups. If the `-rdfg` argument is not specified then the action is applied to all RDF groups. Use the following syntax to set the naming:

```
symcg -cg CgName [-i Interval] [-c Count]
   set <[-name name] [-recovery_rdfg rag]>
     [-rdfg <symid:<rag[,rag,...]|all>[,...] | name:name[,name]>]
```

This command provides a mechanism to group multiple RA groups within a specified CG to perform operations on all associated devices by specifying the RDF group name.

Composite group creation and output example

The `symcg` command can be used to obtain information about the composite groups visible to your host system. Use the following syntax to list the host-visible composite groups:

```
symcg list
```

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 157.

The following example creates a nonconsistent 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 `symcg list` command returns the following output:

```
symcg list

                        C O M P O S I T E    G R O U P S

                                                Number of          Number of
    Name                  Type    Valid  Symms    RAGs    Devs    BCVs   VDEVs     TGTs

   MyNewCG                RDF1    Yes        1       2      32       0       0        0
```

Use the `show` action to return additional details about the `MyNewCG` composite group.

Composite group lists

Use the following syntax to list the available composite groups:

```
symcg [-i Interval] [-c Count]
   list [-offline] [-v] [-apidb | -rdf_consistency]
```

If `-apidb` is specified, only those composite groups that are not in registered with the RDF daemon will be listed. When the `-rdf_consistency` parameter is specified, only those devices that exist in the RDF daemon are listed. The following is an example list operation:

```
symcg list

                        C O M P O S I T E    G R O U P S

                                                Number of              Number of
    Name                  Type    Valid  Symms    RAGs     Devs    BCVs   VDEVs   TGTs

    symcgtest2            REGULAR Yes        1       1        1       2       0       2
    test_cg               RDF1    Yes        1       2        1       0       0       4
    r1                    RDF1    Yes        1       1        0       1       0       1
    test_lock             RDF1    Yes        1       1        0       0       2       2
```

Use the `show` action to return additional details about a composite group. For example, to show the details of composite group `clonetgtcg`, enter:

```
symcg show clonetgtcg

Composite Group Name:  clonetgtcg
  Composite Group Type                              : RDF1
  Valid                                             : Yes
  CG in PowerPath                                   : No
  CG in GNS                                         : Yes
  RDF Consistency Protection Allowed                : No
  RDF Consistency Mode                              : NONE
  Concurrent RDF                                    : No
  Number of RDF (RA) Groups                         :    1
  Number of STD Devices                             :    1
  Number of CRDF STD Devices                        :    0
  Number of BCV's (Locally-associated)              :    0
  Number of TGT's (Locally-associated)              :    1
  Number of VDEV's (Locally-associated)             :    0
  Number of RVDEV's (Remotely-associated VDEV)      :    0
  Number of RBCV's (Remotely-associated STD-RDF)    :    0
  Number of BRBCV's (Remotely-associated BCV-RDF)   :    0
  Number of RRBCV's (Remotely-associated RBCV)      :    0
  Number of RTGT's(Remotely-associated)             :    1
  Number of Symmetrix Units (1):
    {
    1) Symmetrix ID                                  : 000187940256
       Microcode Version                             : 5671
```

```
        Number of STD Devices                          :    1
        Number of CRDF STD Devices                     :    0
        Number of BCV's (Locally-associated)           :    0
        Number of TGT's (Locally-associated)           :    1
        Number of VDEV's (Locally-associated)          :    0
        Number of RVDEV's (Remotely-associated VDEV)   :    0
        Number of RBCV's (Remotely-associated STD_RDF) :    0
        Number of BRBCV's (Remotely-associated BCV-RDF):    0
        Number of RRBCV's (Remotely-associated RBCV)   :    0
        Number of RTGT's (Remotely-associated TGT-RDF) :    1

        Number of RDF (RA) Groups (1):
          {
          1) RDF (RA) Group Number  :  9            (08)
             Remote Symmetrix ID    : N/A
             Microcode Version      : N/A
             Recovery RA Group      : N/A             (N/A)
             RA Group Name          : N/A
             STD Devices (1):
               {

-------------------------------------------------------------------------
                                            Sym  Device            Flags     Cap
             LdevName    PdevName           Dev  Config       Sts  CSR      (MB)

-------------------------------------------------------------------------
             DEV001      N/A                0314 RDF1+Mir     NR   .--      1031
              }
         Remote Target Devices (1):
              {

-------------------------------------------------------------------------
                                            Sym  Device            Flags     Cap
             LdevName    PdevName           Dev  Config       Sts  CSR      (MB)

-------------------------------------------------------------------------
             RTGT001     N/A                0212 STD          RW   .--      1031
              }

          }
        Target Devices (Locally-associated non-RDF) (1):
          {

-------------------------------------------------------------------------
                                            Sym  Device            Flags     Cap
          LdevName    PdevName              Dev  Config       Sts  CSR      (MB)

-------------------------------------------------------------------------
          TGT001      /dev/rdsk/c6t0d76s2    0248 STD          RW   ---      1031
           }
      }
Legend:
  RDFA Flags:
       C(onsistency)    : X = Enabled, . = Disabled, - = N/A
       (RDFA) S(tatus)  : A = Active, I = Inactive, - = N/A
       R(DFA Mode)      : S = Single-session mode, M = MSC mode, - = N/A
```

Changing the state of devices in a composite group

The following commands provide the ability to change a devices state to write-disable, read/write enable, ready, not-ready, or relabel a device state. Use the following syntax to change the state of the devices in a specific composite group:

```
symcg -cg CgName [-i Interval] [-c Count]
   [-noprompt] [-v] [-force]
   [-bcv | -vdev | -tgt] [-star] [-sid SymmID]

   write_disable [LdevName [LdevName...]]

   rw_enable [LdevName [LdevName...]]

   relabel [LdevName [LdevName...]]

symcg -cg CgName [-noprompt] [-v] [-force]
     [-bcv [-hop2] | -rbcv | -brbcv | -rrbcv |
      -vdev [-hop2] | -rvdev | -tgt [-hop2] | -rtgt ]
     [-star]  [-sid SymmID | -rdfg
     <symid:GrpNum[,GrpNum,...]|all [,...] |
      name:RDFGroupName[,RDFGroupName,...]>]

   ready [LdevName [LdevName...]]

   not_ready [LdevName [LdevName...]]

   hold [LdevName [LdevName...]]

   unhold [LdevName [LdevName...]]
```

By default, this command, with no options specified, will write-disable all devices in the STD device list. Use the options provided to limit the action to a specific set of devices. Specify one or a list of logical device names (*LdevName*) to perform the action on a limited number of devices within the CG.

7 Group Name Services

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

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.

Note: If User Authorization is enabled, configuration and management of GNS now require a minimum role of StorageAdmin.

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.

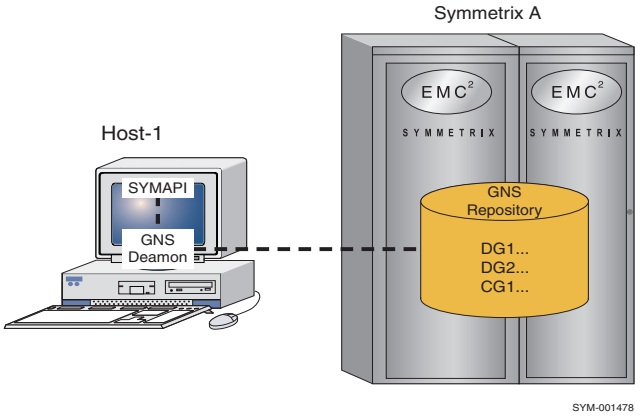


Figure 8 GNS daemon controls GNS activity

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 shown in Figure 8. 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 161.

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 9 on page 154, 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).

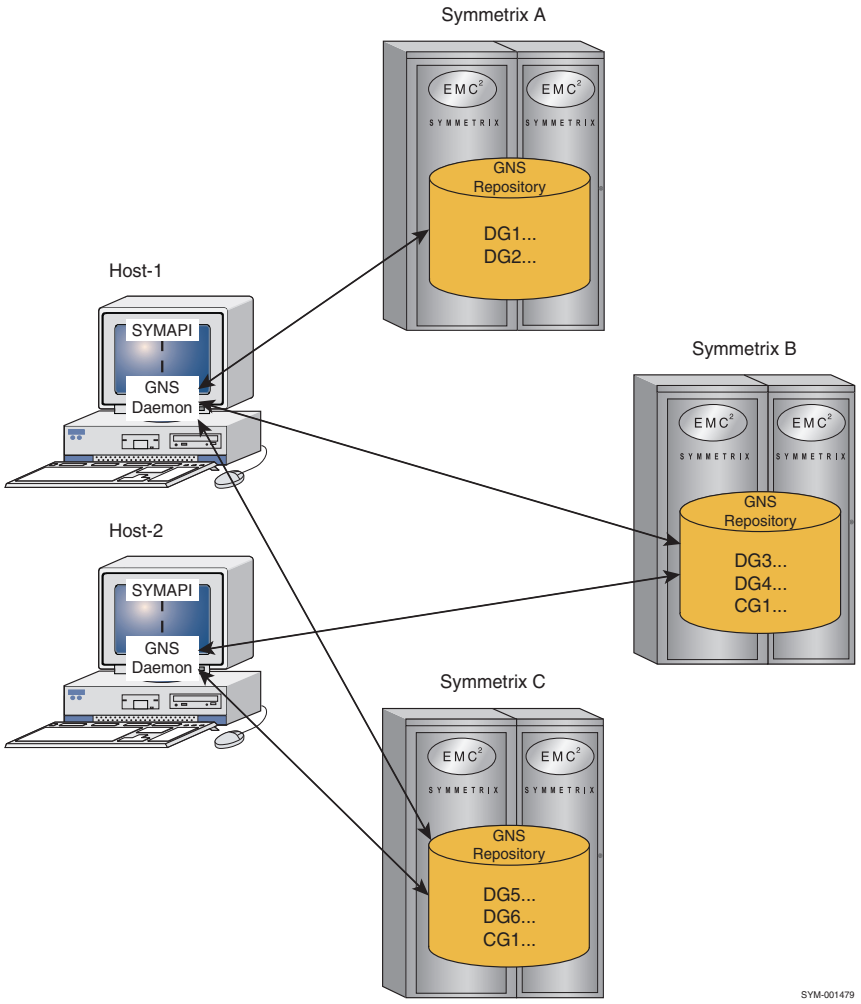


Figure 9 Host-visible GNS state

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

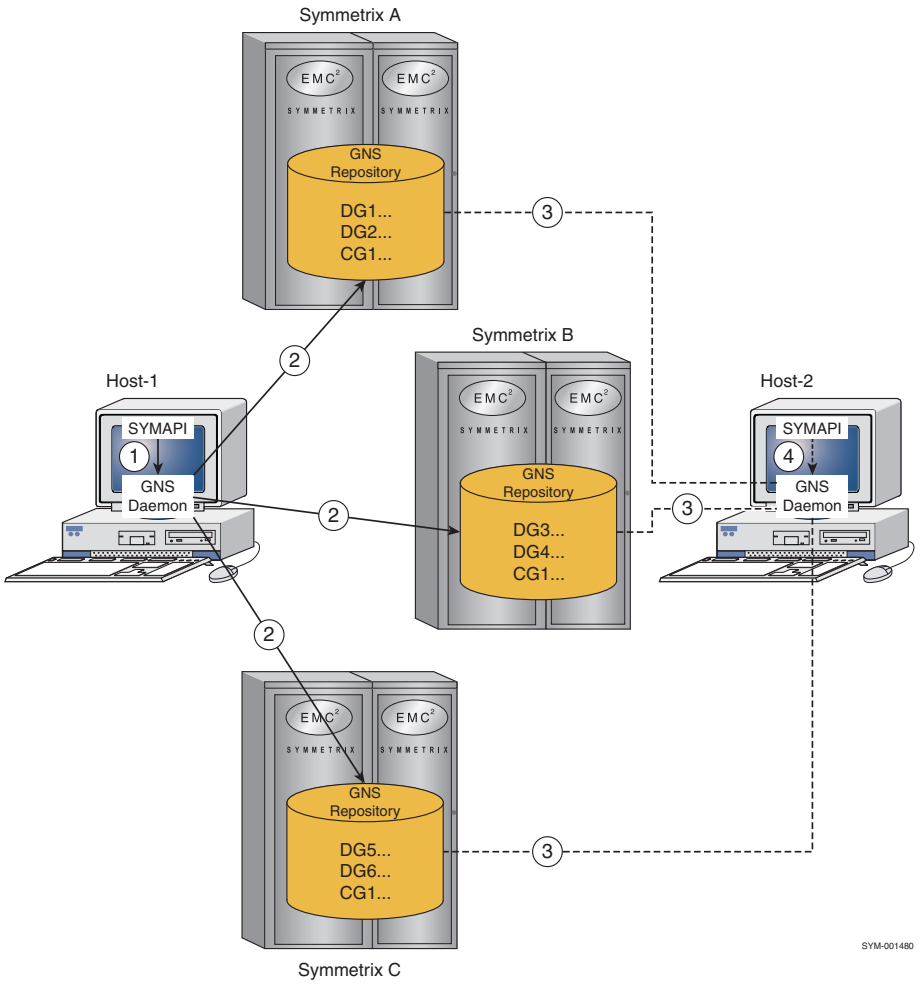


Figure 10 GNS across multiple hosts

GNS and consistency groups

Composite groups can be defined to workMultisession Consistency (MSC, which uses the RDF daemon). In a GNS-enabled environment, certain changes made to any composite group marked for consistency are automatically propagated to the RDF daemon on all relevant hosts, such as changes to the group type, RDF group name, recovery RA group number, device membership, and device LdevName. When updates are made to the GNS repository, the GNS daemon will update the RDF daemon.

For more information on marking composite groups for MSC, refer to "Creating composite groups with multi session consistency (MSC)" on page 142.

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

GNS device groups and SRDF

In an RDF scenario, a local GNS-backed device group (either RDF1 or RDF2) can be automatically mirrored on the remote Symmetrix array via SRDF links—for activation and use during a disaster fail-over situation. By default, the GNS device group definitions are stored on the local (directly attached) Symmetrix—where the local devices are located.

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 restarted 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 164.

GNS behavior in client/server mode

In client/server mode the SYMAPI server's SYMAPI options file setting SYMAPI_USE_GNS must be enabled to use GNS.

Active vs. inactive group lists

When GNS is enabled, group definitions are stored in the global GNS repository in addition to individual host-based configuration database files. 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 multihost environment

When configuring GNS in a multihost 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 invalid to hosts that are not attached to all referenced arrays.

When ready, enable GNS on a single array being sure to configure any options or user authentication that you desire. For details on managing GNS use via the options file, refer to the *Solutions Enabler SYMCLI Command Reference HTML Help*. In addition to enabling GNS on the host, you can configure: GNS User Authentication (set up 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	`/var/symapi/config/daemon_users`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_users`

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 MSC. If you define SRDF consistency groups in a GNS-enabled environment, the GNS daemon running on all relevant hosts will automatically update its 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 22 for details.

Alternatively, the daemon can be started manually via the `stordaemon` command line utility as follows:

```
stordaemon start storgnsd
```

In addition, the daemon can be set to start automatically every time the local host is booted using the following command line:

```
stordaemon install storgnsd -autostart
```

Prestarting 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 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 161). 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 `stordaemon` utility. These control options include:

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

For details, refer to the `stordaemon` man page in the *EMC Solutions Enabler Installation Guide*.

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.

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:

```
stordaemon 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 daemon options file contains a set of parameters that can be modified to affect GNS behavior when using 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 `stordaemon setoption` CLI utility, which programmatically makes changes to the `daemon_options` file. The following is the command-line syntax for this option:

```
stordaemon 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:

```
stordaemon setoption storgnsd -name autorestart=enable
```

The following example shows how this utility can be used to remove an option setting:

```
stordaemon 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

Table 9 lists the possible optional parameter entries for the options file:

Table 9 GNS daemon optional behavior parameters (page 1 of 2)

Optional behavior parameter	= <*OptValue* | *DefaultValue*>	Description
storgnsd:autorestart	= disable | enable	If set to `disable`, the watchdog mechanism is turned off and the daemon will no longer automatically restart if it crashes.
storgnsd:gns_device_poll_interval	= *nnn* | 15	The interval, in seconds, at which to poll locally-attached Symmetrix arrays for group changes.
storgnsd:gns_device_remote_poll_interval	= *nnn* | 60	The interval, in seconds, at which to poll remotely-attached Symmetrix arrays for group changes.
storgnsd:gns_init_remote_symms	= disable | enable	By default (enable), GNS will initialize the GNS group repository on remote Symmetrix arrays while remotely mirroring groups (when the gns_remote_mirror 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 (gns_remote_mirror). • 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 (gns_shadow_file). 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 log file(s) are allowed to grow before they begin to overwrite themselves.
storgnsd:gns_ppath_poll_interval	= *nnn* | 60	The interval, in seconds, at which to check whether PowerPath definitions need to be updated.

Table 9 GNS daemon optional behavior parameters (page 2 of 2)

Optional behavior parameter	= <*OptValue* | *DefaultValue*>	Description
storgnsd:gns_remote_mirror	= enable | disable	If set to `enable`, the daemon will attempt to mirror RDF device group definitions on remote Symmetrix arrays. For more information, refer to "How to mirror remote device group definitions" on page 164.
storgnsd:gns_rmtarr_update_interval	*nn* | 60	The interval, in seconds, at which to check for device group (DG) changes that need to be propagated to remote Symmetrix arrays. Refer to "How to mirror remote device group definitions" on page 164.
storgnsd:gns_shadow_file	= enable | disable | needed	If set to `enable`, the daemon automatically maintains backup copies of GNS data for locally attached Symmetrix arrays. These backups, maintained on the local disk, are automatically restored to the GNS repository on an array if the daemon detects that a repository has been erased (for example, in the event of a power failure under Enginuity 5x67). If set to `needed`, backups are only maintained for Symmetrix arrays with Enginuity versions lower than 5x71. Refer to "GNS backup and restore mechanism" on page 168.
storgnsd:gns_symavoid	= *symid*,*symid*,... |	A comma-separated list of 12-digit Symmetrix IDs for arrays that the daemon should ignore. If this attribute is defined, the daemon will neither look for nor place GNS data on these arrays.
storgnsd:gns_symdiscover_interval	= *nnn* | 600	The interval, in seconds, at which to check for Symmetrix configuration changes.
storgnsd:gns_syminclude	= *symid*,*symid*,... |	A comma-separated list of 12-digit Symmetrix IDs for arrays that the daemon should manage. If this attribute is defined, *only* 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.

Mirroring exceptions

Any group containing one or more of the following configurations cannot be mirrored:

- Groups containing any concurrent devices
- Groups containing any cascaded devices
- Groups containing any hop-2 devices (including 2TGT, 2BCV, 2VDEV deivces)
- If a standard device in the group is paried with an remote RDF BCV
- If there are no standard RDF devices in the group

Impact of remotely mirroring group definitions

The following changes are made to the remote group's definition:

- The remote group definition maintains the same name and type as the local group.
- The group subtype 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.

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 BCV*nnn* are automatically changed to RBCV*nnn*.

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 RBCV*nnn* are automatically changed to BCV*nnn*.

Similarly, the RRBCV logical list becomes the BRBCV list in the remote mirror. Logical names that have a default value of RRBCV*nnn* are automatically changed to BRBCV*nnn*.

As expected, the BRBCV logical list becomes the RRBCV list in the remote mirror. Logical names that have a default value of BRBCV*nnn* are automatically changed to RRBCV*nnn*.

These names are the default names (logical device names) supplied by SYMCLI if the user does not provide one. If the user has overridden the defaults and supplied their own names that look like the above, they will still be transformed as described.

In the unlikely scenario that the user has explicitly supplied aliases, for example, BCV*nnn* and RBCV*nnn*, 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.

Figure 11 on page 166 illustrates the mirrored group data:

```
--- Local DG Group, on Symm1 ---
Type             RDF1
Role             Have a Remote Mirror
Remote Symm      Sym2
Local RDFGrp     2
Remote RDFGrp    5
Devices          0001,0002,0003
Remote-Devs      0091,0092,0093
LDevNames    DEV001,DEV002,DEV003

BCVs             0010,0011
Remote-BCVs      0020,0021,0022
BCV LdevName     BCV001,BCV002
RBCV LdevName    RBCV001,RBCV002,RBCV003

Vendor           EMC
ModTime          12345
GateKeepers      /dev/rdsk/c2t0d2
```

```
--- Remote DG Group, on Symm2 ---
Type          RDF2
Role          Am a Remote Mirror
Remote Symm   Sym1
Local raGrp   5
Remote raGrp  2
Devices       0091,0092,0093
Remote-Devs   0001,0002,0003
Dev Aliases   DEV001,DEV002,DEV003

BCVs          0020,0021,0022
Remote-BCVs   0010,0011
BCV Aliases   BCV001,BCV002,BCV003
RBCV Aliases RBCV001,RBCV002

Vendor        EMC
ModTime       12345
```

Figure 11 Mirrored remote group definition

Identify GNS groups

After you have configured your Symmetrix environment to support the GNS option, you can begin to create and modify groups. Use `symdg show` *DgName*, for device groups, or `symcg show` *CgName*, for composite groups, to determine if a group is stored in the GNS group list.

Group output

For a device group, the output of the command `symdg show` *DgName* identifies whether a group definition is stored in GNS. The sample output below shows that group `MyRegDeviceGroup` is stored in the GNS list since the output parameter **`Device Group in GNS`** has a value of **`Yes`**. It also shows the device group's GNS mirror state.

```
% symdg show MyRegDeviceGroup

Group Name: MyRegDeviceGroup

    Group Type                                  : REGULAR
    Device Group in GNS                         : Yes (Is Mirror)
    Valid                                       : Yes
    Symmetrix ID                                : 000184500160
    Group Creation Time                         : Tue Aug 4 16:44:18 2006
    Vendor ID                                   : EMC Corp
    Application ID                              : SYMCLI

    Number of STD Devices in Group              :   20
    Number of Associated GKs                    :    0
    Number of Locally-associated BCVs           :   20
    Number of Locally-associated VDEVs          :    0
. . .
```

In addition, for a composite group, the output of the command `symcg show` *CgName* identifies whether a group definition is stored in GNS. The sample output below shows that group `MyCompGroup` is stored in the GNS list since the output parameter **`CG in GNS`** has a value of **`Yes`**. It also shows the composite group's GNS mirror state. The group definition has been enabled for RDF consistency via MSC.

```
% symcg show MyCompGroup

Composite Group Name: MyCompGroup

  Composite Group Type                          : RDF1
  Valid                                         : Yes
  CG in PowerPath                               : No
  CG in GNS                                     : Yes (Is Mirror)
  RDF Consistency Protection Allowed            : Yes
  RDF Consistency Enabled                       : Yes

  Number of RDF (RA) Groups                     :    4
  Number of STD Devices                         :   64
  Number of BCVs (Locally-associated)           :    0
  Number of VDEVs (Locally-associated)          :    0
  Number of RBCVs (Remotely-associated STD-RDF)  :    0
  Number of BRBCVs (Remotely-associated BCV-RDF) :    0
  Number of RRBCVs (Remotely-associated RBCV)    :    0

. . . .
```

GNS backup and restore mechanism

GNS group definitions are stored within each individual Symmetrix array. A GNS daemon can be configured to automatically back up and restore GNS data from individual arrays. When a 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 is stored in the following directory on the host machine:

UNIX `/var/symapi/gns/SID.shadow`

Win32 `c:\\Program Files\\EMC\\SYMAPI\\gns\\SID.shadow`

Note: Because the maintenance of numerous backup images can impact performance, it is recommended you do not enable it on all GNS daemons. Typically, it is recommended that you configure a maximum of two or three 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 161.

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, be aware of the possibility of group name collision and invalid groups and what SYMAPI does to rectify each scenario.

Group name collisions

Each GNS daemon attempts to ensure that group names are unique (relative to a group type). It is possible, however, for duplicate-named groups to occur.

For example, it is possible for two different groups to be created with the same name on different Symmetrix arrays. A subsequent consideration change could then allow the GNS daemon to see both groups.

If a GNS daemon detects duplicate group names, it returns a state flag informing clients of this fact and modifies the group names to be unique by attaching a numeric suffix. For example, duplicate device groups with the name `MyDG` could be modified in the following way:

```
MyDG#12345678
MyDG#87654321
```

While in this state (duplicate names), the only operations permitted are those used to resolve the name conflict: group rename and delete (with the `-force` flag). For example:

```
symdg delete MyDG#12345678 -force
symdg rename MyDG#87654321 MyDG
```

Note: For information on renaming composite groups, see "Renaming a composite group" on page 147.

If users at two different hosts simultaneously create groups of the same type with the same name using devices on different Symmetrix arrays, 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.

Invalid groups

A composite group is marked as invalid 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 invalid.

Groups are also placed into an invalid 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 invalid.

This could happen if one of the arrays is reinitialized 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 from that host. A user there would have to use the `-force` flag to delete the group. Groups marked as invalid can be deleted with the `-force` flag.

8 Statistics

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

The Symmetrix array maintains statistical counters for performance-critical operations. Using the `symstat` command, you can view the performance statistics derived from these counters.

The statistics command (`symstat`) performs the following:

- Queries Symmetrix devices to capture raw performance counts and store them in memory.
- Retrieves the performance counts for the Symmetrix array as a whole.
- Retrieves the performance counts for a director or director port.
- Retrieves the performance counts for one or more Symmetrix devices.
- Retrieves the performance counts for one or more Symmetrix device groups, composite groups, or RDF groups.
- Retrieves the performance counts for a selection of, or all, Symmetrix disks.
- Retrieves the timestamp of the performance count sample.
- Retrieves and displays replication session statistics for SRDF/A.
- Retrieves Gig-E iSCSI network statistics.

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 devices 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, back-end 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 tracks 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 173.

Performance type options

The `symstat` command provides different types of performance display options in Table 10:

Table 10 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.
ISCSI	Reports Gig-E network statistics.
CACHE	Reports cache activity for selected front-end or remote link directors, or SRDF/A sessions.
MEMIO	Reports cache memory to disk activity for selected devices.
PATH	Reports R-Copy path information for nonincremental sessions. Symmetrix arrays that have all or some incremental sessions will report an error.
CYCLE	Reports 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).
RDF	Reports SRDF statistics from the perspective of RA groups, devices, or directors.

REQUESTS statistics

Using `symstat`, you can display performance statistics such as I/O requests and throughput for the devices within a Symmetrix array, a device group, or SRDF/A sessions by specifying the REQUESTS type. REQUESTS is the default type for `symstat` operations, therefore, if no type is specified, REQUESTS performance information is returned.

Note: When using Solutions Enabler V7.0 `symstat` command against Symmetrix arrays running Enginuity 5874, some device and director ouput fields have been changed to support 64-bit counters. Refer to the"Compatibility mode: Preserving old Solutions Enabler scripts" on page 20 for information on returning output as seen in prior Solutions Enabler versions.

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. The following example returns performance statistics for Symmetrix array 150:

```
symstat -type REQUEST -sid 150 -i 10 -c 100


DEVICE            IO/sec                  KB/sec                % Hits   %Seq     Num WP
09:30:11       READ    WRITE         READ          WRITE         RD  WRT  RD  WRT  Tracks
   1     2     3       4             5             6               7      8        9

09:30:21 002B 1345344 18900230 1345344836009850 1345344836009850 100 99   99   99    240
```

The following statistics are returned:

1. Host timestamp of the sample.
2. Active Symmetrix device name and host physical device name (if the device is visible).
3. Read cache request rate (requests per second).
4. Write cache request rate (requests per second).
5. Read throughput rate (kilobytes per second).
6. Write throughput rate (kilobytes per second).
7. Read/write cache hit ratio (percentage of write hits).
8. Sequential read/write ratio (percentage of sequential reads).
9. Write-pending track count.

Device statistics

Using `symstat` with the REQUESTS type (default) specified, you can display performance statistics such as I/O requests via SAs (or other front-end adapters) and throughput on a device.

From the device list shown in the device group statistical run, select a logical device to examine I/O requests and throughput activity and choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of 5 seconds and a sample count of 3 on logical device DEV001 in device group prod_r1, enter:

```
symstat -i 5 -c 3 -g prod_r1 -ld DEV001
```

The fields returned are identical to those returned in "REQUESTS statistics" on page 174, except that the statistics are limited to the one logical device specified.

SRDF/A session statistics

Using symstat with the REQUESTS type (default) specified, you can display a table of host writes and SRDF throughput information for SRDF/A sessions in Symmetrix arrays by specifying the -repType rdfa option.

When requesting SRDF/A session statistics for a specified Symmetrix array, the output differs depending upon the flag arguments specified:

- If the -sid flag is used with no further qualifiers ([-g ALL | -rdfg <#|ALL> | -cg <*CgName*|ALL>]), then the output display lists SRDF/A sessions that have a status of active.
- If the -sid flag is used with qualifier -g all, then the output display list all device groups, regardless of status. If the -sid flag is used with the qualifier -rdfg ALL, then the output display lists all RDF groups, regardless of status. If the -sid flag is used with the qualifier -cg ALL, then the output display lists all composite groups, regardless of status.
- If the -g *DgName* device group qualifier is used (whereby the -sid option is not specified), then the output display only lists details for the specified device group.
- If the -sid flag is used with the qualifier -rdfg #, then the output display lists just the specified RDF group.
- If the -sid flag is used with the qualifier -cg *CgName*, then the output display lists just the specified composite group.
- If the -v option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: 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:

Note: 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:

```
symstat -i 30 -c 3 -sid 0011 -RepType rdfa

SRDF/A Session Throughput Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:23:30

                                    Host Write Workload  RDF Session Throughput
                        RA   Session ------------------- ----------------------
Device Group Name       Grp  Number    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

                                  Host Write Workload  RDF Session Throughput
                     RA   Session ------------------- ----------------------
Device Group Name    Grp  Number    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

                                  Host Write Workload  RDF Session Throughput
                     RA   Session ------------------- ----------------------
Device Group Name    Grp  Number    IO/sec    MB/sec      IO/sec      MB/sec
-----------------  ---  -------  --------  --------- ----------  ----------
RaGrpNum_25         25       24       343       2.7         549         5.7
RaGrpNum_26         26       25       519       4.1         531         6.0
RaGrpNum_27         27       26       398       3.1         741         7.9
```

The following statistics are returned:

1. Name of the device group.
2. The RA Group number.
3. SRDF/A session number.
4. Number of host I/O writes per second.
5. Workload rate, in MB per second of host writes.
6. Number of SRDF I/O writes per second in the SRDF/A session.
7. Throughput rate, in MB per second of the total I/O writes in the SRDF/A session.

Example: Verbose SRDF/A session statistics

For example, to return verbose details about SRDF/A session statistics on Symmetrix array `6163` for all active device groups, with a sample intervals of `30` seconds and a sample count of `3`, enter:

```
symstat -sid 6163 -i 30 -c 3 -reptype rdfa -v

SRDF/A Session Information:

Symmetrix Id                                    : 000000006163
Timestamp                                       : 16:26:24

System Write Pending Limit                      : 236462   (7.22 GB)
Cache Slots available for all SRDF/A sessions   : 222274   (6.78 GB)
Total Local Write Pending Count                 : 0
Total System Write Pending Count                : 0

  Device Group Name                             : RaGrpNum_60
  RA Group Number                               : 60
  Session Type                                  : RDF1
  Session Number                                : 59
  Session Priority                              : 33
  Number of Device in Session                   : 3
  Session Status                                : Active
  Session Attributes                            : MSC
  Local Active Cycle Number                     : 443
```

```
Minimum Cycle Time                          : 00:00:30
Average Cycle Time                          : 00:00:30
Last Cycle Time                             : 00:00:30
Time Since Last Cycle Switch                : 00:00:14
Time that R2 is behind R1 (RPO)             : 00:00:44
Tracks not Committed to the R2 Side         : 0
RDF1 Active (Capture) Cycle Size            : 0
RDF1 Inactive (Transmit) Cycle Size         : 0
RDF2 Active (Receive) Cycle Size            : N/A
RDF2 Inactive (Apply) Cycle Size            : N/A
Cache Slots in Use                          : 0
Percent Available Cache Used                : 0.0
Host Write Workload (IO/s)                  : 0
Host Write Workload (MB/s)                  : 0.0
RDF Throughput R1->R2 (IO/s)                : 0
RDF Throughput R1->R2 (MB/s)                : 0.0
RDF Throughput R2->R1 (IO/s)                : 0
RDF Throughput R2->R1 (MB/s)                : 0.0
Last Restore Time    (R2 only)              : 00:00:00
Average Restore Time (R2 only)              : 00:00:00
Total HA writes                             : 0
Total HA repeat writes                      : 0
```

R-Copy session statistics

Using `symstat` with the REQUESTS type (default) specified, you can display a table of host writes and SRDF throughput information for Open Replicator R-Copy sessions in Symmetrix arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name <SessionName>]
```

To return R-Copy session statistics for a Symmetrix array 0003, enter:

```
symstat -sid 0003 -reptype rcopy -i 30 -c 3
```

```
                                      RCopy
                                  ---------------
          Device                       KB/sec
15:28:45                          PULL      PUSH

15:29:20  0644  (Not Visible )    15189         0
          0645  (Not Visible )    15276         0
          0646  (Not Visible )    15275         0
          0647  (Not Visible )    15383         0
```

This example shows the KB/sec throughput for an R-Copy, hot-pull operation.

Mainframe (Ficon) statistics

Using `symstat` with the REQUESTS type (default) specified, you can display a table of mainframe Ficon throughput information in Symmetrix arrays by specifying the `-EF` option. You can either specify a specific port number or all port numbers, as shown in the example below.

To return mainframe Ficon statistics, enter:

```
symstat -sid 179 -i 5 -c 2 -EF all -type REQUESTS
```

```
          DIRECTOR     IO/sec        KB/sec         Avg. Queue
12:07:50                          READ    WRITE       Length

12:07:56 EF-16B           122   12345    67890          15

12:08:01 EF-16B           126   34521    78900          12
```

This example shows the KB/sec READ (bytes received) and WRITE (bytes transmitted), which includes both protocol overhead and data.

BACKEND statistics and PREFETCH activity

Statistics for I/O requests on back-end directors, the throughput on the corresponding devices, and prefetch disk activity on back-end directors can be returned using the Solutions Enabler SYMCLI.

Returning BACKEND statistics

Using `symstat` with the BACKEND type specified, you can display performance statistics such as I/O requests via a DA (back-end director) and throughput on a device.

From the device list shown in the device group statistical run, select the logical device for which you want to examine I/O requests and throughput activity. Choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of `5` seconds and a sample count of `3` on logical device `DEV001` in device group `prod_r1`, enter:

```
symstat -type BACKEND -i 5 -c 3 -g prod_r1 -ld DEV001
```

The display would be similar to the front-end REQUESTS display showing read/write I/O, prefetched tracks, and unused track activity.

PREFETCH activity

From the Symmetrix list, select the Symmetrix array that you want to examine for prefetch disk activity on the back-end directors (DAs). Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of `5` seconds and a sample count of `4` for prefetch activity on Symmetrix `010050010333`, enter:

```
symstat -i 5 -c 4 -sid 333 -da all -type prefetch
```

The following is an example output:

```
symstat -i 5 -c 4 -sid 333 -da all -type prefetch


          DIRECTOR   Preftch       Preftch Trk/sec
16:42:30             Trk/sec       Used  Unused
   (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 timestamp of sample.
2. Active director name type and number.
3. Track prefetch rate (tracks per second).
4. Used prefetched track rate (tracks per second).
5. Unused (superseded) prefetched track rate (tracks per second).

Director statistics

Using `symstat`, you can display the I/O requests and throughput activity performance statistics on any or all directors. In addition, you can return director statistics for R-Copy sessions. To return a list of directors available on all visible Symmetrix arrays, use `symcfg list -dir all`.

Director I/O requests

From the list, select the Symmetrix array for which you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run.

Note: When using Solutions Enabler V7.0 `symstat` command against Symmetrix arrays running Enginuity 5874, some device and director ouput fields have been changed to support 64-bit counters. Refer to the"Compatibility mode: Preserving old Solutions Enabler scripts" on page 20 for information on returning output as seen in prior Solutions Enabler versions.

For example, to output statistics for all directors on Symmetrix array 150, enter:

```
symstat -type REQUESTS -sid 150 -dir ALL -i 10 -c 100
```

```
  DIRECTOR         IO/sec                        Cache Requests/sec                 % RW
09:32:13                         READ              WRITE            RW            Hits
   ❶         ❷      ❸            ❹                   ❺               ❻               ❼

09:32:18 FA-7E 29224856000 10003424332159670 1000000234654674 11003424566814344 99
```

The following statistics are returned:

1. Host timestamp of sample.
2. Active director identifier and number.
3. I/O rates (operations per second) for:
   - Host to Front-end directors
   - Disk to Back-end directors
   - Local to Remote link directors
4. Read cache request rate (requests per second).
5. Write cache request rate (requests per second).
6. Total read/write cache request rate (requests per second).
7. Cache hit ratio (percentage of cache read and write hits).

R-Copy session statistics

Using `symstat` with the REQUESTS type (default) specified, you can display a table of directors on the specified array and SRDF throughput information for Open Replicator R-Copy sessions in Symmetrix arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name SessionName]
```

To return R-Copy session statistics for a Symmetrix array 097, enter:

```
symstat -sid 097 -reptype rcopy -dir 7a -i 5

                      RCopy      Cache Requests/sec     %RW
16:06:49   Director   MB/sec     READ   WRITE   RW      Hits
   1          2          3        4      5      6       7
16:07:04   FA-7A          58      716      0    716     80
                        ------   ------ ------ ------   ---
                           58      716      0    716     79
```

The following statistics are returned:

1. Host timestamp of sample.
2. Active director identifier and number.
3. R-Copy director throughput in MB/sec.
4. Number, or the count, of cache read requests per second.
5. Number, or the count, of write requests per second.
6. Total number, or count, of combined read and write requests per second.
7. Cache hit ratio (percentage of cache read and write hits).

PORT statistics

Using `symstat` with the PORT type specified, you can display performance statistics such as I/O requests and throughput of director ports for a specific Symmetrix array.

How to return director port statistics

Choose a Symmetrix array for which you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run. The `symstat -type PORT` syntax allows you to return statistics for one or all directors on one or all Symmetrix arrays. In addition, you can optionally specify one or all port numbers. Use the following syntax for director port statistics:

```
symstat [-i Interval] [-c Count] -type PORT [-sid SymmID]
   -dir <#|ALL> [-port <#|ALL>]
```

For example, for sample intervals of `5` seconds, and a sample count of `3` (for I/O requests and throughput) on Symmetrix `0256`, enter (the output follows):

```
symstat -i 5 -c 5 -type PORT -sid 0256 -dir ALL

15:24:30    DIRECTOR    PORT      IO/sec     Kbytes/sec
15:24:36      FA-15C       0          40              0
15:24:41      FA-15C       0          28              0
15:24:47      FA-15C       0          27              0

15:24:52    DIRECTOR    PORT      IO/sec     Kbytes/sec
15:24:52      FA-15C       0          24              0
```

R-Copy session director port statistics

Using `symstat` with the PORT type specified and the `-reptype` set to `rcopy`, you can display a table of director ports on the specified array and session throughput information for Open Replicator R-Copy sessions.

```
symstat -sid 0003 -reptype rcopy -type port -dir 14c -i 30 -c 3

                                  RCopy
            Director   ------------------------
15:34:07      Port     MB/sec  Ceiling  Devices
  (1)         (2)      (3)      (4)      (5)
            FA-14C:1   23        30        4

            FA-14C:1   24        30        4
```

The following statistics are returned:

1. Host timestamp of sample.
2. Active director and port number.
3. R-Copy session throughput in MB/sec.
4. The maximum percentage of that director/port's bandwidth that is allotted to R-Copy operations.
5. Number of remote devices. The number of remote devices is only available when all R-Copy sessions are nonincremental.

ISCSI statistics (Gig-E)

Using `symstat` with the iSCSI type specified, you can display Gig-E performance statistics, such as I/O requests and throughput of director ports for a specific Symmetrix array. Statistics can be returned for Gig-E directors configured as RDF directors (`-RE`) and for Gig-E directors (`-SE`) configured as SCSI directors.

TCP Connection-based statistics

To return TCP connection-based statistics, specify the `-tcp_conn` option and the SCSI director number (ALL can be specified as a value). For example, enter:

```
symstat -type ISCSI -i 5 -c 5 -sid 237 -tcp_conn -SE 20

GigE ISCSI Statistics
TCP Connection Based Statistics
                        IP Addr:Port                    Queue    Segments/sec   RTT
15:40:57        Local               Remote          Send Recv  Trans   Recv
15:40:58 192.168.100.101:123  192.168.100.211:456  100  200   1000   4000    10

15:41:03 192.168.100.101:123  192.168.100.211:456  200  300   2000   3000    20
```

Network Interface-wide statistics

To return network interface-wide statistics, specify the `-gige_dir` option and the SCSI director number (ALL can be specified as a value). For example, enter:

```
symstat -type ISCSI -i 5 -c 5 -sid 237 -gige_dir -SE 20

GigE ISCSI Statistics
Network Interface Wide Statistics
              IP Pkts       ICMP Msgs       TCP Sgmts       UDP Pkts
              /sec          /sec            /sec            /sec
15:41:10   Sent    Rcvd   Sent    Rcvd    Sent    Rcvd   Sent    Rcvd
15:41:11   1000    2000   1500    1200    3000    2500   1800    2200

15:41:16   2000    1500   1800    1400    2000    2300   2000    2000
```

RDF speedlimit statistics

To return RDF speedlimit statistics, specify the `-rdf_spdlmt` option and the RDF director number (ALL can be specified as a value). For example, enter:

```
symstat -type ISCSI -i 5 -c 5 -sid 237 -rdf_spdlmt -RE 24

GigE ISCSI Statistics
RDF Speedlimit Statistics

15:41:33 10

15:41:38 15
```

RDF network compression statistics

To return RDF network compression statistics, specify the `-rdf_nw_comp` option, RDF director number (ALL can be specified as a value), and the IP address of the remote array. For example, enter:

```
symstat -type ISCSI -i 5 -c 5 -sid 237 -rdf_nw_comp -RE 24 -ip 192.168.100.150

GigE ISCSI Statistics
RDF Network Compression Statistics
                Compressed Bytes/sec          Uncompressed Bytes/sec
15:41:45     Transmitted     Received     Transmitted       Received
15:41:46        12345678     11222456        34456788       45667899

15:41:51        14566778     13456456        45636788       56757899
```

Speedlimit group statistics

To return RDF speedlimit group statistics, specify the `-spdlmt_grp` option and the RDF director number (ALL can be specified as a value). For example, enter:

```
symstat -type ISCSI -i 5 -c 5 -sid 237 -spdlmt_grp -RE 24

GigE ISCSI Statistics
Speedlimit Group Statistics
              Speedlimit Grp   Director   Active   Speedlimit    Bandwidth
15:42:09                       Number             State
15:42:10                         0x1        16C       10           20          1000

15:42:15                         0x2         1D       12           25          2000
```

MEMIO statistics

Using `symstat`, you can display performance statistics, such as cache memory to disk I/O on a device group. However, you may need to first 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.

Note: When using Solutions Enabler V7.0 `symstat` command against Symmetrix arrays running Enginuity 5874, some device and director ouput fields have been changed to support 64-bit counters. Refer to the "Compatibility mode: Preserving old Solutions Enabler scripts" on page 20 for information on returning output as seen in prior Solutions Enabler versions.

Example: Memory and I/O statistics

The following display is an example output of cache to disk activity for the devices of Symmetrix array 150:

```
symstat -type MEMIO -sid 150 -i 10 -c 100

DEVICE                                    Tracks/sec                 % Dev
09:31:24      WP Tracks        Prefchd           Destgd          WPmax
    (1)           (2)             (3)               (4)            (5)

09:31:29 002B   119003    347825900002675400  37232619000345011   84
```

If `-output` was specified, the above output would apprear in the following XML format:

```
<SymCLI_ML>
  <Statistics>
    <time>Thu Nov 13 2008 06:00:42</time>
    <Memory>
      <dev_name>002B</dev_name>
      <pd_name>/dev/sdaa</pd_name>
      <wp_tracks>119003</wp_tracks>
      <pf_tracks>347825900002675400</pf_tracks>
      <destaged_tracks>37232619000345011</destaged_tracks>
      <wp_max_pct>84</wp_max_pct>
    </Memory>
 <SymCLI_ML>
```

The following statistics are returned:

1. Host timestamp of sample and Symmetrix device name.
2. Write pending track count not yet destaged to disk.
3. Track prefetch rate (tracks per second).
4. Track destage rate (tracks per second).
5. Write-pending device limit percentage (percentage of device parameter).

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.

DMSP statistics

Using `symstat` with the DMSP type specified, you can display DMSP (dynamic mirroring service policy) statistics such as service policy and I/O performance attributes for a specified Symmetrix device.

Note: DMSP statistics are not supported on Enginuity version 5772 or later.

Device dynamic mirroring service policy statistics

From the device list shown in the device group statistical run, select the logical device for which you want to examine DMSP statistics and throughput activity. Choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of `5` seconds and a sample count of `3` on logical device `DEV001` in device group `prod_r1`, enter:

```
symstat -type DMSP -i 5 -c 3 -g prod_r1 -ld DEV001
```

The display provides DMSP information for mirror 1 and 2, the preferred mirror, new and old mirror service policies, the percent of sequential I/O, the percent of write I/O, and the activity level.

Disk statistics

Using `symstat` with the DISK type specified, you can display performance statistics, such as I/O requests and throughput on a physical disk, at specified time intervals for the duration of a specified count.

Specifying a disk

Select the physical DA director number, interface ID, and disk SCSI ID interface on a specified Symmetrix array for which you want to examine I/O requests and throughput activity. To retrieve the statistics of all DA directors, interfaces, or disk SCSI IDs, use the value ALL. For example, for sample intervals of `5` seconds and a sample count of `3` on disk `02A:C5`, enter:

```
symstat -type disk -i 5 -c 3 -sid 577 -disk 2a,C,5
```

If you want to retrieve statistics at the same interval and sample count, but for all SCSI IDs on DA 02A, interface C, enter:

```
symstat -type disk -i 5 -c 3 -sid 577 -disk 2a,C,ALL
```

The display provides the disk ID, I/O read/writes, and kilobytes per read and write for each disk.

CACHE statistics

You can return data about the cache activity on the front-end directors, LRU, or SRDF/A sessions.

Front-end director statistics

Statistics for front-end directors can be returned by specifying options for SA, EA, FA, FE, CA, RA, director number, or all. In addition you can choose the interval and number of samples (counts) for the statistical run. For example, to return cache activity for all SAs on Symmetrix `010050010333`, for sample intervals of `5` seconds and a sample count of `4`, enter:

```
symstat -i 5 -c 4 -sid 097 -sa all -type cache
```

```
          DIRECTOR    Misses/sec    Disconnects/sec
15:21:52              Cache Read    System    Device
   (1)       (2)          (3)          (4)       (5)
15:22:03 FA-15C             0            24         0
```

As shown in this example, there was one sample of cache activity to report on one director during the sample period and `symstat` lists the following statistical items:

1. Host timestamp 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 write pending limit.

LRU cache management statistics

Using `symstat`, you can display the LRU cache statistics. However, you may need to first list the LRUs and Symmetrix arrays (use `symcfg list -lru ALL`) to see the LRUs for all units and specify a Symmetrix ID.

Note: LRU statistics are not supported on Enginuity version 5*x*70 or later.

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 timestamp 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 nonzero 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 you request SRDF/A session statistics for a specified Symmetrix array, the output differs depending on the flag arguments specified:

- If the -sid flag is used with no further qualifiers ([-g ALL | -rdfg <#|ALL> | -cg <*CgName*|ALL>]), the output displays SRDF/A sessions that have a status of active.
- If the -sid flag is used with qualifier -g all, the output displays all device groups, regardless of status. If the -sid flag is used with the qualifier -rdfg ALL, the output displays all RDF groups, regardless of status. If the -sid flag is used with the qualifier -cg ALL, the output displays all composite groups, regardless of status.
- If the -g *DgName* device group qualifier is used (whereby the -sid option is not specified), then the output displays only details for the specified device group.
- If the -sid flag is used with the qualifier -rdfg #, then the output displays just the specified RDF group.
- If the -sid flag is used with the qualifier -cg *CgName*, then the output displays just the specified composite group.
- If the -v option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: R1 SRDF/A session statistics for active groups

For example to return SRDF/A session statistics for all active device groups on Symmetrix array `0011`, enter:

```
symstat -type cache -i 30 -c 3 -sid 0011 -repType rdfa
```

The following statistics are returned:

```
symstat -type CACHE -RepType rdfa -sid 16

SRDF/A Session Cache Summary Information

Symmetrix Id                                    : 000190300016
Timestamp                                       : 06:20:44

System Write Pending Limit                      : 364376   (11.12 GB)
Cache Slots available for all SRDF/A sessions   : 342513   (10.45 GB)
Total Local Write Pending Count                 : 0
Total System Write Pending Count                : 0

                                Session        Rdfa DSE           Cache    Cache
                     RA  ---------------------- ---------------    Slots    Full
Device Group Name    Grp Type Num Pri Status    Thr   UsedTrks     In Use    (%)
-------------------- --- ---- --- --- -------- --- ----------- ----------- -----
RaGrpNum_63           63 RDF1  62  33 Active    60           0           0   0.0
RaGrpNum_05            5 RDF1   4  33 Active    60           0           0   0.0

Total                                             ----------- ----------- -----
  Slots                                                     0           0   0.0
  GB                                                     0.00        0.00
```

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 an SRDF/A cycle. Typically, this limit is approximately 80 percent of the total cache available, however, in some case it is less.
- **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 percent 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.
- **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 an SRDF/A device, or both. When a cache slot has both local and SRDF/A writes pending flag set, it is only counted once.
- **Rdfa DSE** — Identifies the SRDF/A DSE threshold and used track counts.

- **Cache Slots in Use** — Cache slots in use is the total number of cache slots in the active and inactive cycle for that SRDF/A session. These represent any cache slot with a SRDF/A write pending.

  Note: The sum of the Cache Slots in Use and the Local Write Pending Count does not equal the Total System Write Pending Count. This is because Cache Slots in Use and Local Write Pending Count include every cache slot in those states, while the Total System Write Pending Count includes slots with either of those states, but not both. So a cache slot with both a local write pending and an SRDF/A write pending would increase the individual counts of each, but only increment the total count by one.

- **Cache Full (%)** — The percentage of the available SRDF/A cache in use for the device group.

PATH statistics

Using the SYMCLI you can report R-Copy path information for nonincremental sessions. Symmetrix arrays that have all or some incremental sessions will report an error.

Note: These statistics are only available when all R-Copy sessions are nonincremental.

The data can either be presented by session, where the data is organized by session name (`-by_session` option), or by port, where the data will be organized by director port (`-by_port` option).

Organized by session

The following example shows the R-Copy PATH statistics for Symmetrix array 003 organized by session name:

```
symstat -sid 0003 -reptype rcopy -type path -by_session -i 30 -c 3

Time       Session  Control      Target           Tracks      Director  RCopy
Stamp       Name    Device     Destination     Total Remaining   Port   Ceiling

15:43:24  hotpull   0644       N/A              264960       0 FA-14C:1    30
          hotpull   0645       N/A              264960       0 FA-14C:1    30
          hotpull   0646       N/A              264960       0 FA-14C:1    30
          hotpull   0647       N/A              264960       0 FA-14C:1    30

Time       Session  Control      Target           Tracks      Director  RCopy
Stamp       Name    Device     Destination     Total Remaining   Port   Ceiling

15:44:00  hotpull   0644       N/A              264960       0 FA-14C:1    30
          hotpull   0645       N/A              264960       0 FA-14C:1    30
          hotpull   0646       N/A              264960       0 FA-14C:1    30
          hotpull   0647       N/A              264960       0 FA-14C:1    30

Time       Session  Control      Target           Tracks      Director  RCopy
Stamp       Name    Device     Destination     Total Remaining   Port   Ceiling

15:44:36  hotpull   0644       N/A              264960       0 FA-14C:1    30
          hotpull   0645       N/A              264960       0 FA-14C:1    30
          hotpull   0646       N/A              264960       0 FA-14C:1    30
          hotpull   0647       N/A              264960       0 FA-14C:1    30
```

Organized by port

The following example shows the R-Copy PATH statistics for Symmetrix array 003 organized by director port:

```
symstat -sid 0003 -reptype rcopy -type path -by_port -i 30 -c 3

Time      Director   RCopy    Port   RCopy   Control      Tracks       Session
Stamp       Port    Ceiling  KB/sec  KB/sec  Device   Total Remaining   Name

15:44:44  FA-14C:1     30         -       -    0644    264960       0  hotpull
                                              0645    264960       0  hotpull
                                              0646    264960       0  hotpull
                                              0647    264960       0  hotpull


Time      Director   RCopy    Port   RCopy   Control      Tracks       Session
Stamp       Port    Ceiling  KB/sec  KB/sec  Device   Total Remaining   Name

15:45:20  FA-14C:1     30        17       0    0644    264960       0  hotpull
                                              0645    264960       0  hotpull
                                              0646    264960       0  hotpull
                                              0647    264960       0  hotpull


Time      Director   RCopy    Port   RCopy   Control      Tracks       Session
Stamp       Port    Ceiling  KB/sec  KB/sec  Device   Total Remaining   Name

15:45:55  FA-14C:1     30        19       0    0644    264960       0  hotpull
                                              0645    264960       0  hotpull
                                              0646    264960       0  hotpull
                                              0647    264960       0  hotpull
```

CYCLE statistics

The CYCLE type statistics return a table of cycle information for device groups (SRDF/A sessions) in Symmetrix arrays that have active status or have nonzero cache usage.

When you request SRDF/A session statistics for a specified Symmetrix array, the output differs depending on the flag arguments specified:

- If the `-sid` flag is used with no further qualifiers (`-g all`, `-rdfg # | ALL`), the output displays SRDF/A sessions that have a status of `active`.
- If the `-sid` flag is used with qualifier `-g all`, the output displays all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg all`, the output displays all RDF groups, regardless of status.
- If the `-sid` flag is used with the qualifier `-rdfg #`, the output displays just the specified RDF group.
- If the `-g` *`DgName`* device group qualifier is used (without specifying the `-sid`), the output displays only details for the specified device group.
- If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: All device groups

For example, to return SRDF/A session cycle statistics of active sessions for Symmetrix array `0011`, enter:

`symstat -type cycle -i 30 -c 3 -sid 0011 -RepType rdfa -g all`

The following statistics are returned for one count:

```
SRDF/A Session Cycle Summary Information
Symmetrix Id   : 000187400011
Timestamp      : 14:40:16

                                  Session         Cycle Time  (sec)     Cycle Size
                           ------------------ -------------------- ---------------
                        RA              Active               Last
Device Group Name Grp  Type Number Cycle#  Min Avg Last Switch  Active Inactive
  (1)              (2)  (3)   (4)    (5)   (6) (7) (8)   (9)     (10)    (11)
----------------- ---  ---- ------ ------ --- --- ---- ------  ------ --------
< . . . >
RaGrpNum_25         25  RDF1     24     171  15  19   20      4     732      526
RaGrpNum_26         26  RDF1     25     165  15  19   22      2     591     4221
RaGrpNum_14         14  RDF1     13     173  15  19   19      4       0        0
RaGrpNum_16         16  RDF1     15     173  15  19   20      4       0        0
RaGrpNum_18         18  RDF1     17     172  15  19   19      4       0        0
RaGrpNum_19         19  RDF1     18     177  15  19   19      4       0        0
RaGrpNum_20         20  RDF1     19     171  15  19   19      4       0        0
RaGrpNum_27         27  RDF1     26     165  15  19   21      2     426     1294

Legend for the Attribute of Cycle Size:
  RDF1: Active = Capture   Inactive = Transmit
  RDF2: Active = Receive   Inactive = Apply
```

Note: 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).

RDF statistics

The RDF type statistics extend the statistical information provided by the RDF director to external applications providing greater visibility into the performance and behavior of Symmetrix arrays in the field. These statistics can be used to monitor on-going activity and to analyze problematic behavior.

SRDF/A session statistics by RA group

Statistics for SRDF/A sessions can be monitored by RA group number. To return statistics for all RA group numbers, specify `all`.

```
symstat -sid 179 -i 5 -c 2 -type RDF -rdfg 11

RDF Group Level I/O Statistics:

          GRP         IO/sec          MB/sec       % Hits      IO Svc Time       Q
13:33:31          READ    WRITE     READ    WRITE       RD       Min      Max     Len
13:33:31  11       100      200    12345    34500       30        23        0      10

13:33:37  11       110      220    15000    45000       40        30        0      15
```

SRDF device-level statistics

SRDF statistics can be monitored at the device level. Device-level statistics can be obtained by specifying a Symmetrix device name (`-dev` *SymDevname*), logical device name (`-ld` *LDevname*), or device group name (`-g` *DgName*).

Note: A device group name (`-g` *DgName*) must be specified when a logical device name (`-ld` *LDevname*) is used.

For example, to return SRDF device-level statistics for device 37, enter:

```
symstat -sid 179 -i 5 -c 2 -type RDF -dev 37

RDF Device Level I/O Statistics:

           DEV   RDF          IO/sec          MB/sec      % Hits      IO Svc Time
13:37:31       GRP MODE    READ  WRITE     READ  WRITE      RD       Min      Max
13:37:33   37  17    0     100    200     12345  34500      30        30        0

13:37:38   37  17    0     110    220     15000  45000      40        50        0
```

SRDF statistics by director number

SRDF statistics can be monitored at the RDF director level. Director-level statistics can be obtained by specifying a specific director number or specifying `all`.

For example, to return SRDF director level statistics, enter:

```
symstat -sid 179 -i 5 -c 2 -type RDF -dir 1d

RDF Director Level I/O Statistics:

          DIR        IO/sec          MB/sec       % Hits      IO Svc Time       Q
14:08:33           READ   WRITE     READ   WRITE      RD      Min      Max     Len
14:08:34  32        100     200    12345   34500      30       30        0      10
14:08:34  40        120     220    24567   55432      25       40        0      12

14:08:40  32        110     210    15000   45000      40       50        0      15
14:08:40  40        130     230    26265   61250      35       55        0      20
```

SRDF link-level statistics

SRDF director-level statistics can be further enhanced to return thoughput information of the RDF links. To return link-level details, specify the `-rdflink` option as shown in the example below.

```
symstat -sid 179 -i 5 -c 2 -type RDF -dir 2c -rdflink

RDF Link Level I/O Statistics:

          LINK       IO/sec          MB/sec       % Hits      IO Svc Time     Echo
14:18:27           READ  WRITE     READ   WRITE      RD      Min      Max     Delay
14:18:29    0       100    200    12000   21000      35       30        0        50
14:18:29    1       110    210    13000   22000      40       40        0        60
14:18:29    2       120    220    14000   23000      45       50        0        70
14:18:29    3       130    230    15000   24000      50       60        0        80

14:18:35    0       110    210    23000   31000      45       35        0        60
14:18:35    1       120    220    24000   32000      50       45        0        70
14:18:35    2       130    230    25000   33000      55       55        0        80
14:18:35    3       140    240    26000   34000      60       65        0        90
```

9

Change Tracker

This chapter introduces the Change Tracker SYMCLI (`symchg`) command and explains how to monitor Symmetrix data objects for change within your storage environment.

Introduction to Change Tracker

The EMC Solutions Enabler Change Tracker software is used to measure changes to data on a Symmetrix volume or group of volumes. The Change Tracker command (`symchg`) is used on FBA devices to mark selected areas of Symmetrix disk storage so that Symmetrix logical volume occupying those areas can be monitored for change.

Change Tracker uses DeltaMark bitmap technology (formerly called the Symmetrix Differential Data Facility) to identify logical blocks that have been changed on a Symmetrix FBA device. Before change tracking can begin, a Change Tracker (DeltaMark) session must be created using the `symchg create` command. The `symchg mark` command is then used to perform a timestamp and mark the selected area of disk storage occupied by a data object using a DeltaMark bitmap.

This allows a marked area to be examined at a later time to view changes to the stored object. The tracked changes can also be saved to a log file for later analysis. Change Tracker data is often used to analyze and design TimeFinder and Symmetrix Remote Data Facility (SRDF) configurations.

The Symmetrix system stores information about track changes internally, and makes this information available to applications through a DeltaMark bitmap vector or table. Each DeltaMark bitmap contains a bit for every track on the monitored volume. When a write to the volume alters a track of information, the bitmap is updated to reflect that the track has been changed. (Subsequent changes to that same track have no effect on the bitmap.) Figure 12 illustrates a DeltaMark bitmap table.

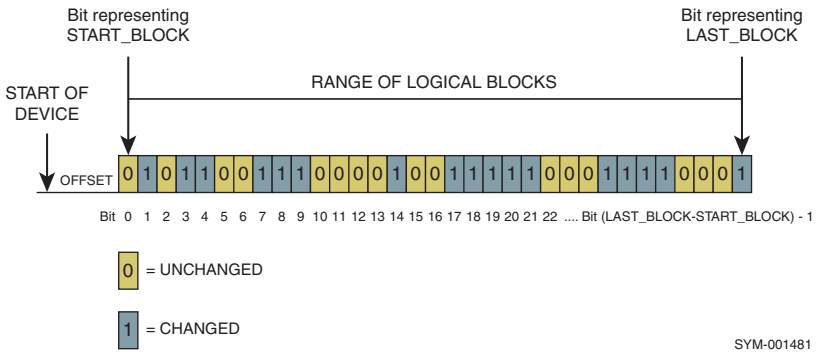


Figure 12 A DeltaMark bitmap table

Practical uses

Using TimeFinder and SRDF software to mirror data from a source Symmetrix system to a remotely associated Symmetrix system located across the hall or across the globe creates a global data storage solution. That is, production data can be available at a remote site for data mining, e-business content delivery, application development, testing, backups, and numerous other uses. Moreover, SRDF configurations provide a recovery solution for component or site failures between remotely mirrored devices.

With the wide acceptance of SRDF for remote storage and as a disaster tolerance and disaster restart tool, assessing incremental changes to data over discrete periods of time is a necessary part of planning for data transfer and storage at remote sites. Change Tracker assists in this task by measuring volume changes, providing information so that informed decisions can be made about the communication line bandwidths and links required for such configurations.

Change Tracker command summary

The Change Tracker command (`symchg`) allows you to monitor selected Symmetrix disk storage objects using DeltaMark bitmap technology to track changes and generate reports for later analysis. Symmetrix objects that can be monitored include individual devices, standard or BCV devices listed in a device group or composite group, and devices listed in a device file.

Table 11 summarizes the SYMCLI `symchg` commands you can use to create a Symmetrix Change Tracker session, mark areas to be monitored, remove or delete a session, and generate a report.

Table 11 Change Tracker command summary

Command	Argument	Displays
`symchg`	`create`	Creates a Change Tracker session for the disk device containing the disk storage object to be monitored.
	`delete`	Unmarks the disk storage object and deletes the Change Tracker session.
	`list`	Lists all disk storage objects that are currently marked or all devices that have a Change Tracker session created.
	`mark`	Marks disk storage objects as unchanged so that new changes can be tracked. Requires that a Change Tracker session first be created.
	`remove`	Unmarks the disk storage object while retaining the Change Tracker session.
	`report`	Generates a report on the amount of change for the specified object in a given log file. Reports can be generated for device groups, composite groups, and device files.
	`view`	Displays the amount and rate of change for marked disk storage objects or shows if a Change Tracker session exists for a specified disk storage object.

For information about performing Change Tracker control operations, refer to "Monitoring data objects for changes" on page 202. For details about the `symchg` command syntax and other SYMCLI commands, refer to the *Solutions Enabler SYMCLI Command Reference HTML Help*.

Note: Change Tracker control operations using the `symchg` command require that a DeltaMark license key is installed on your system. Refer to the *EMC Solutions Enabler Installation Guide* for license information.

Monitoring data objects for changes

To mark areas of Symmetrix disk storage for change monitoring, you first create a Change Tracker session for the objects that you want to monitor. Creating a Change Tracker session creates DeltaMark bitmaps for the specified objects. Once a session has been established, the objects can be marked at any particular point in time, which resets the bitmaps for those objects (that is, marks all tracks as unchanged). Marking the objects is the starting point from which changes will be measured.

Monitoring the objects is accomplished by viewing the objects in a session. Viewing objects processes the bitmaps that correspond to the objects and displays the amount and rate of change. Viewing can be performed at discrete intervals of time and for a particular number of executions, and the bitmaps can also be reset after each sample. The display may be saved to a user-defined log file, where it is stored in a comma-separated format, enabling the data to be imported into a spreadsheet for analysis. A summary report can be produced from the results in a log file, and the log file can be used to perform capacity planning by specifying particular variables and calculating other parameters.

The `symchg` command operates on Symmetrix disk storage objects by specifying a Symmetrix device name, a logical device name, a device group, a composite group, or a device file. Individual objects may be removed from a session without terminating that session (so other objects in the same session continue to be monitored), and a session can be deleted, terminating the monitoring of all objects included in that session.

Note: SYMCLI `symchg` only monitors FBA devices. CKD devices can be monitored using the mainframe change tracker utility.

Creating a Change Tracker session

Before marking objects, you must create a Change Tracker session for the device(s) containing the objects by using the `symchg create` command.

The `symchg create` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName create
symchg -g DgName create
symchg -cg CgName create
symchg -file FileName create
```

For example, to create a Change Tracker session for a single Symmetrix device (`17C`) on Symmetrix ID 000125600055, enter:

```
symchg create dev 17c -sid 55
```

To create a Change Tracker session for devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB create
```

To create a Change Tracker session for devices listed in a device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 create
```

You can mark and examine more than one object per session and you can specify a Symmetrix device name (*SymDevName*) or a logical device (*LdevName*) name within a

device group or composite group for the session. You can specify the standard or BCV device (`-bcv`) of a device group or composite group.

Creating a device text file

You can create a Change Tracker session for devices listed in a device text file. The device file syntax contains a single column of devices specified by Symmetrix device name (*SymDevName*). Lines in the device file that begin with a pound symbol (#) will be ignored. The device filename (`-file` *FileName*) will be inserted into the command line for control operations. A sample of a device text file listing Symmetrix devices is shown in the following example.

Example of a device text file

```
# dev_file_1
# Symmetrix Device List for Change Tracker session
#
00A
00B
00C
057
058
# End
```

Marking a data object

After creating a session, you can mark the device storage objects for change monitoring using the `symchg mark` command. This allows objects to be marked from a particular point in time, with all tracks being marked as unchanged. Once marked, the objects can be monitored for changes.

The `symchg mark` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName mark
symchg -g DgName mark
symchg -cg CgName mark
symchg -file FileName mark
```

Note: You must use the same convention for the `mark` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

When you mark an object, an entry is added or updated for the specified object in the `symchg` database. The mark command causes the DeltaMark bitmaps to be reset (no changes) for the devices of the specified object.

You can specify a Symmetrix device name (00A) or a logical device name (DEV001) for the object to be marked. You can also mark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to mark Symmetrix device (`17C`) on Symmetrix ID 000125600055, enter:

```
symchg mark dev 17c -sid 55
```

To mark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB mark -bcv
```

For example, to mark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 mark
```

Listing marked objects

You can use the `symchg list` command to list disk storage objects that have been marked.

For example, to list all logical disk storage objects that are currently marked, enter:

```
symchg list
```

To list all disk devices for Change Tracker sessions that have been created, enter:

```
symchg list -session
```

To list all disk devices that are visible to the host for which Change Tracker sessions that have been created, enter:

```
symchg list -session dev
```

The following is an example of the `symchg list` command output for Symmetrix 000125600055.

```
symchg -sid 55 -session list
```

```
Symmetrix ID: 000125600055

        C H A N G E   T R A C K E R   D E L T A M A R K   S E S S I O N S

          Device Name          Directors                    Device
-------------------------------------  ------------------  ----------------------------------------------------------
                                                                                                          Cap
Physical                        Sym  SA :P  DA  :IT  Config        Attribute     Sts                    (MB)
-------------------------------------  ------------------  ----------------------------------------------------------
/dev/rdsk/c3tl3d9s2             108  14A:1  02B:D3  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d10s2            109  14A:1  01A:C3  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d11s2            10A  14A:1  01B:D2  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d12s2            10B  14A:1  02A:C2  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d13s2            10C  14A:1  02B:D1  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d14s2            10D  14A:1  01B:D1  2-Way Mir     Grp'd         RW                     4096
/dev/rdsk/c3tl3d15s2            10E  14A:1  02A:D1  2-Way Mir     Grp'd         RW                     4096
```

Removing marked objects

You can unmark a disk storage object from the `symchg` database without terminating the Change Tracker session by using the `symchg remove` command.

The `symchg remove` control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName remove
symchg -g DgName remove
symchg -cg CgName remove
symchg -file FileName remove
```

Note: You must use the same convention for the `remove` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

The remove command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object, while maintaining the session.

You can specify a Symmetrix device name (i.e. 00A) or a logical device name (i.e. DEV001) for the object to be unmarked. You can also unmark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to unmark Symmetrix device (17C) on Symmetrix ID 000125600055, enter:

```
symchg remove dev 17c -sid 55
```

To unmark BCV devices in the device group ProdDB, enter:

```
symchg -g ProdDB remove -bcv
```

To unmark devices listed in the device text file dev_file_1, enter:

```
symchg -file dev_file_1 remove
```

Deleting marked objects

You can unmark a disk storage object from the symchg database and delete (terminate) the Change Tracker session by using the symchg delete command.

The symchg delete control operation can be performed on a single Symmetrix device (*SymDevName*), or by device group, composite group, or device text file:

```
symchg SymDevName delete
symchg -g DgName delete
symchg -cg CgName delete
symchg -file FileName delete
```

Note: You must use the same convention for the delete command as was used with the symchg create command when the session was created. If the session was created using a device group, composite group, or device file, the devices must be managed using the same method.

The delete command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object and deletes the session.

You can specify a Symmetrix device name (i.e. 00A) or a logical device name (i.e. DEV001) for the object to be unmarked. You can also unmark the BCV devices (-bcv option) of a device group or composite group.

For example, to unmark Symmetrix device (17C) on Symmetrix ID 000125600055 and delete the session, enter:

```
symchg delete dev 17c -sid 55
```

To unmark BCV devices in the device group ProdDB, enter:

```
symchg -g ProdDB delete -bcv
```

To unmark devices listed in the device text file dev_file_1, enter:

```
symchg -file dev_file_1 delete
```

Viewing marked objects

You can process the DeltaMark bitmap corresponding to the disk storage object to display the amount and the rate of change by using the `symchg view` command. For example, to display the amount and the rate of change for Symmetrix device (`17C`) on Symmetrix ID 000125600055, enter:

```
symchg view dev 17c -sid 55
```

Note: You can verify the existence of a Change Tracker session for a specified object by including the `-session` option.

To display the amount and the rate of change for BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB view -bcv
```

To display the amount and the rate of change for devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 view
```

Change information for marked objects in a device group, composite group, or listed in a device file can be saved to a separate log file. The data written in log file is based on 32k track size; optionally, you can choose to return the output in the native track size by specifying the `-native` option instead. For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB`, enter:

```
symchg -g ProdDB view -reset -log /tmp/logfile -c 10
```

The `-reset` option resets the DeltaMark bitmaps for each reporting interval and requires using the count (`-c`) or interval (`-i`) options. The reset after each measurement of changed tracks is called the DELTA collection method.

The SUM collection method calculates the cumulative number of changed tracks from the starting point of when the object was first marked. To collect changed tracks for both the SUM and DELTA collection methods at the same time, you can use the `-multi` option.

For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB` using both the SUM and DELTA collection methods, enter.

```
symchg -g ProdDB view -multi -log /tmp/logfile -c 10
```

Note: At each specified interval, the log file will capture both the cumulative number of changed tracks and the number of changed tracks per interval.

Generating reports of changes

Change Tracker allows you to generate two kinds of reports—SUM and DELTA. Reports can be generated for devices in a device group, composite group, or devices listed in a device file.

SUM reports

SUM reports track changes from the point-in-time the DeltaMark bitmap was marked at the beginning of change measurement. Subsequent writes to the same track are not monitored, because the bitmap is not reset after every measurement interval. For instance if the initial *mark* action occurred at 6 A.M. and change data was collected each hour for 12 hours, it would be possible to measure the tracks changed between 6 and 7 A.M., 6 and 8 A.M., 6 and 9 A.M. and so on. It would not be possible to measure the changes between 7 and 8, or 8 and 9 using the SUM report. The SUM reports lend themselves to refresh interval estimations for SRDF/AR installations. Understanding the change measurement at various intervals, enables one to determine the amount of SRDF bandwidth required for different SRDF/AR refresh intervals. Refer to the log file entry below.

```
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 59525 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 91946 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM

           A      B          C         D    E      F                      G      H
                I
```

The fields in the log file entry are:

A - Date and time that the sample was collected.

B - Symmetrix ID.

C - Type of entity (device group).

D - Name of the device group.

E - Symmetrix device name.

F - Physical device name. If the device has not been host visible, this field would say `Not Visible`.

G - Number of tracks changed since the group was marked.

H - Total number of tracks on device.

I - SUM indicates that the number in column G represents the cumulative tracks changed since the group was marked.

DELTA reports

DELTA reports track changes that have occurred since the last measurement interval. The DeltaMark bitmap is reset after each measurement of changed tracks. Thus, an hourly report from 6 A.M. to 6 P.M. using the DELTA method would identify how

many tracks changed during any given one-hour period. This is useful for determining peak change rates for a given interval of data collection.

To assess the bandwidth requirements for an SRDF/AR implementation, you should first collect a SUM report to determine how often you would want to refresh the remote site. Once the determination of the refresh interval has been finalized (T hours), generate a DELTA report using T hours as the collection interval. This will result in identifying the peak number of changed tracks during any T-hour interval. Refer to the log file entry below for an example of a DELTA report.

```
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 16 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 8 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 803 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 549 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 24 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 48 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 1085 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 809 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 25, 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 13, 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 561 , 131070 , DELTA
04/10/2006 06:10:21 , 000184600266 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 541, 131070 , DELTA
```

Viewing reports

When viewing the changes for an object of a device group, the output can either be sent to `stdout` or saved to a user-defined log file. The information saved to the log file is stored in a comma-separated format. This allows the data to be imported into a spreadsheet for analysis. The results in the log file can be examined with the following command:

```
symchg [-ra NumRAs] [-rate #] [-resync [mmmm|hh:mm]]
        [-start mmddyyyy hh:mm] [-stop mmddyyyy hh:mm]
        [-backend] [-range StartDev:EndDev] [-native] [-v]

        report -log LogFile <-g DgName | -cg CgName> [-bcv]

        report -file DevFile -log LogFile [-sid SymmID]
```

Each report in a log file shows the amount of change for a specified object of a device group. For more information about the syntax of `symchg`, refer to the *Solutions Enabler SYMCLI Command Reference HTML Help*.

Report options

Table 12 lists the command line options available for generating a report using the `symchg report` command:

Table 12 SYMCLI report options (page 1 of 2)

Option	Description
-backend	Displays information about the DA and back-end disk address of marked devices.
-bcv	Generates a report for the BCV devices in the device group or composite group.
-cg	Specifies a composite group on which to report.
-file	Specifies a file containing a device list with one *SymDevName* per line.
-g	Specifies a device group on which to report.
-log	Specifies the log file (`-log LogFile`) to read from.
-native	Reports data based on native track size.
-sid	Specifies the 12-digit ID of a Symmetrix array.

Table 12 SYMCLI report options (page 2 of 2)

Option	Description
-start	Defines the start point in the log file to begin processing data. Format is *mmddyyyy hh:mm*. Default is the beginning of the log file.
-stop	Defines the stop point in the log file to stop processing data. Format is *mmddyyyy hh:mm*. Default is the end of the log file.
-ra	Defines the number of RA directors to use when resynchronizing device pairs in the group.
-range	Specifies a range of Symmetrix device names (*SymDevName)* for which to generate the report. Format is *SymDevStart:SymDevEnd*.
-rate	Specifies the transfer rate to use when resynchronizing device pairs in the group. Format is in *Kb/second*.
-resync	Specifies the required time frame in minutes, or hours and minutes, for resynchronizing RDF device pairs in the group. Format is *mmmm* or *hh:mm*.

Example procedure

The following is an example of typical steps to follow when you are saving change information to a user-defined log file and you are generating a report:

1. Create Change Tracker sessions for each standard device in a device group (such as DATABASE_DG), enter:

   ```
   symchg create -g DATABASE_DG
   ```

2. Mark the devices within the device group (DATABASE_DG).

   ```
   symchg mark -g DATABASE_DG
   ```

   This will reset the DeltaMark change information to zero for each device in the device group.

3. Save the change information for the devices of the group (DATABASE_DG) to a log file (`/tmp/change_data.log`) by entering:

   For SUM Reports:

   ```
   symchg view -g DATABASE_DG -log /tmp/change_data.log \ -i 60 -c 100
   ```

   For DELTA Reports:

   ```
   symchg view -g DATABASE_DG -log /tmp/change_data.log \ -reset -i 60
   -c 100
   ```

   Note: In this example, there were 100 samples made at an interval of 60 seconds and the DeltaMark change information is reset after each sample.

   For combined SUM and DELTA Reports:

   ```
   symchg view -g DATABASE_DG -log /tmp/change_data.log -multi -i 60 -c
   100
   ```

4. Produce a summary report of the changed track information for each device in the group (DATABASE_DG) by entering:

   ```
   symchg report -g DATABASE_DG -log /tmp/change_data.log -v
   ```

The data in the log file is in a comma-separated format, so you could also import it into a spreadsheet to view the data.

5. If you need to provide RDF capacity planning that determines how long it would take to resynchronize a device group (DATABASE_DG) with RA directors at a transfer rate of 1000 KB/second, enter:

   ```
   symchg report -g DATABASE_DG -log /tmp/change_data.log -rate 1000
   -ra 2
   ```

6. If you need to view information for a device group (DATABASE_DG) on your screen showing the log entries, enter:

   ```
   symchg view -g DATABASE_DG
   ```

7. To list the objects that are marked and being monitored for change, enter:

   ```
   symchg list
   ```

8. To verify a Change Tracker session has been created for a device group (DATABASE_DG) and that it is marked, enter:

   ```
   symchg view -g DATABASE_DG -session
   ```

9. To remove the marked object (DATABASE_DG), enter:

   ```
   symchg remove -g DATABASE_DG
   ```

   This does not delete the sessions for the devices of the group. It simply records the object indicating it is no longer being monitored (from a host view point).

Using Change Tracker for RDF capacity planning

The Symmetrix Remote Data Facility (SRDF) maintains a mirror image of data at the device level in Symmetrix systems located in physically separate sites. The mirroring in SRDF can be accomplished in real-time (Synchronous mode) or near real-time (Asynchronous or Adaptive Copy modes). These modes of operation are described fully in the *EMC Solutions Enabler Symmetrix SRDF Family Product Guide*. The Adaptive Copy mode is especially useful when transferring large amounts of data during data center migrations, consolidations, and in data mobility environments.

When performing periodic RDF transfers using Adaptive Copy mode, it is important to estimate the rate of change of data on the volumes to be mirrored in order to make informed decisions about communication line parameters. Change Tracker can help in this process by calculating certain RDF capacity values (the resynchronization time, the number of remote adapters (RAs), and the RA transfer rates) based on the amount of change to monitored objects.

In the SRDF configuration shown in Figure 13, the R1-BCV is used to periodically transmit changed tracks from the primary site to the remote site. When the BCV is established with the standard as a BCV pair, the RDF link to its R2 target is suspended. When you split the local BCV pair, you can resume the RDF link, sending the changes to the remote site. Using Change Tracker to measure the extent of changes to the local BCV pair while the R1-BCV is suspended from its R2 target allows you to make informed choices about link bandwidth and frequency of transmission for the RDF transfer.

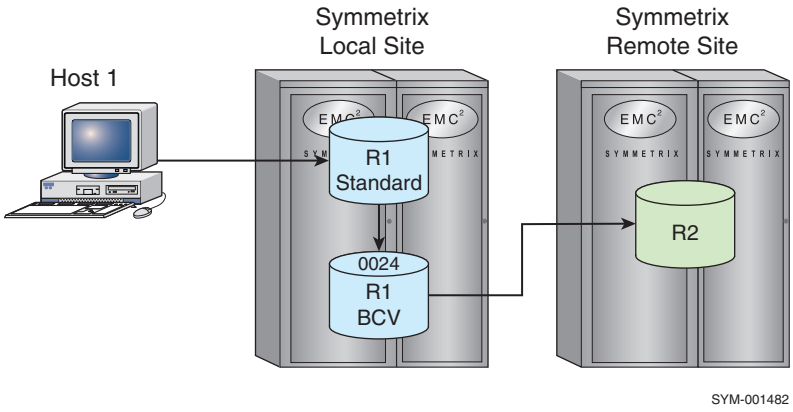


Figure 13 SRDF configuration

The following steps outline an example of setting up devices and using Change Tracker to perform RDF capacity planning:

1. Identify the device group that you want to monitor. If the group does not exist, you can create it using the `symdg` command. For example, to create a device group named deltadg, enter:

   ```
   symdg create deltadg
   ```

2. Add a device (Symmetrix device name 085 on Symmetrix number 000000003264) to the device group:

   ```
   symld -g deltadg -sid 3264 add dev 085
   ```

   By default, SYMCLI assigns a logical device name (of the form DEVnnn) to this device.

3. Create a Change Tracker session for the device group:

   ```
   symchg -g deltadg create
   ```

4. List the sessions that have just been created for the device group:

   ```
   symchg -session list
   ```

5. Mark the device group (this is the starting point from which changes will be measured):

   ```
   symchg -g deltadg mark
   ```

6. In a log file, record cumulative track changes that occur on each device in the device group. This example collects 60 samples at two-minute (120 seconds) intervals in a log file named deltalog:

   ```
   symchg -g deltadg -log deltalog -i 120 -c 60 view
   ```

7. Use the collected data from the log file for RDF capacity planning by providing two of the following three variables to a `symchg report` command:

   - The number of links
   - The link bandwidth
   - The amount of time available to synchronize the source (R1) and target (R2) devices

   The `symchg report` command calculates the third variable.

10

Gatekeeper Devices

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

Overview

Low-level I/O commands executed using SYMCLI are routed to the Symmetrix array by a Symmetrix storage device that is specified as a *gatekeeper*. The gatekeeper device allows SYMCLI commands to retrieve configuration and status information from the Symmetrix array without interfering with normal Symmetrix operations. The gatekeeper must be accessible from the host where the commands are being executed.

Note: If the SYMCLI is running on a service processor, there is only one communication path (gatekeeper). Therefore, there is no need to manage it with `symgate`.

In addition to managing gatekeepers using the SYMCLI, gatekeepers management can also be handled automatically using the Base Daemon. For more information, refer to the *EMC Solutions Enabler Installation Guide*.

Gatekeeper candidates

All devices in your storage array are potential gatekeeper candidates. While you can explicitly dedicate devices as gatekeepers, and even associate those gatekeepers with specific device groups (as described later), SYMCLI can utilize any available device as a gatekeeper.

Note: BCV devices *cannot* be used or defined as gatekeepers. VCMDB devices can be used as a gatekeeper but *cannot* be defined as a gatekeeper.

SYMCLI selects a gatekeeper based upon a preestablished 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 ($\leq$ 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 nonmetadevices 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 six 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, and 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 214.

Note: Choosing to use the Base Daemon to automatically manage gatekeepers will override any actions performed with `symgate`.

Defining a gatekeeper

Use `symgate` to define a device as a gatekeeper device. This command adds the specified physical device name to the list of gatekeeper devices.

For example on UNIX, to define device `c2t0d2` as a gatekeeper device, enter:

```
symgate define pd c2t0d2
```

On Windows, to define device `\\.\physicaldrive2` as a gatekeeper, enter:

```
symgate define pd physicaldrive2
```

SYMCLI uses one of the devices in the list as a gatekeeper, unless you have associated a gatekeeper explicitly with the device group of the device to which SYMCLI is sending a control command.

Note: You must add a device to the gatekeeper device list before you can associate it with a device group.

Undefine a gatekeeper

Use `symgate` to remove a Symmetrix device from the gatekeeper device list.

For example, to remove Symmetrix device `000C` from the gatekeeper device list, enter:

```
symgate undefine dev 000C
```

If the gatekeeper is associated with a device group, the gatekeeper must first be disassociated from the device group using `symgate` (also, refer to "Disassociating from a device group" on page 218).

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
```

11

Events and Logs

This chapter describes how to configure, manage, and query Symmetrix events and logs.

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 the event or error in detail

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 *Solutions Enabler SYMCLI Command Reference HTML Help*.

How to configure log options

Table 13 lists the options to customize how and where events are logged in your Symmetrix environment.

Table 13 Log file configuration options (page 1 of 2)

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:
	`setenv SYMCLI_NOLOGGING 1`
	To turn logging back on, enter:
	`unsetenv SYMCLI_NOLOGGING`

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:
	`setenv SYMCLI_LOG filename`
	To turn daily log files back on, enter:
	`unsetenv SYMCLI_LOG`

Table 13 Log file configuration options (page 2 of 2)

Logging option	Configuration description

Allow Undated SYMAPI Log File Names	You can allow the creation of undated SYMAPI log files by setting the environment variable SYMAPI_DATED_LOGFILE_NAME in the options file to `disable`.
	To allow undated log file names, set:
	`SYMAPI_DATED_LOGFILE_NAME=DISABLE`
	To reenable dates, set:
	`SYMAPI_DATED_LOGFILE_NAME=ENABLE`

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 `FORMAT2`. This formats the date as *yyyy-mm-dd*.
	To change the date format, set:
	`SYMAPI_LOGFILE_DATE_FORMAT=FORMAT2`
	To change the date format back to the default (*mm/dd/yyyy*), set:
	`SYMAPI_LOGFILE_DATE_FORMAT=FORMAT1`

Log File Configuration Options	You can change format of optional fields within each log record by setting the environment variable SYMAPI_LOGFILE_FORMAT in the options file. Zero or more of the following optional fields can be included: • `pid` — include the process ID. • `tid` — include the thread ID. • `userid` — include the user ID (useful with User Authorization is enabled). • `activityid` — include the activity ID. The default value is `pid tid`. To see the log file format, create an entry in the options file with values separated by spaces. For example: `SYMAPI_LOGFILE_FORMAT = userid pid tid activityid`
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: `SYMAPI_LOGFILE_RETENTION=60` **Note:** When the Log File Retention option is set, it overrides all existing Log File Retention values including those set for the service processor during initialization. The default for the service processor is 30 days. Setting this options file value will override that default. In addition, if you have accumulated many years of log files prior to setting this option, when the option is set, it deletes those log files that do not meet the specified criteria.

Monitoring Symmetrix events via SYMCLI

The `symevent` command allows an administrator to monitor or track events within a Symmetrix array that may affect its operation. In some cases, reported events represent conditions that have already been repaired. This command allows the user to monitor the Symmetrix array for all reported events.

You can use `symevent` to either:

- `Monitor` — This action causes the command to run in the foreground, polling the Symmetrix array for new events every interval in seconds, either until the iteration count is satisfied or the program is stopped.
- `List` — This action examines the history of events, stored on the Symmetrix array, for those that meet the requested criteria. The `-start` and `-end` options allow the user to retrieve events that occurred between certain time bounds. In addition, you can sort the returned events based upon the reporting director using the `-DIR` option.

You can further refine your query to a Symmetrix ID and restrict the reported events to those of a minimum severity (warnings, errors, or fatal events).

Monitor

In its most basic invocation, `symevent` polls for and reports on all events, on all locally-connected Symmetrix systems, every 10 seconds, continuously:

```
symevent monitor
```

The following example polls for and displays events with a severity of warning or greater on Symmetrix 0207 every 10 seconds for a 50-second period:

```
symevent monitor -sid 0207 -i 10 -c 50 -warn
Symmetrix ID: 000192600207

Detection time           Dir    Src  Category     Severity     Error Num
------------------------ ------ ---- ------------ ------------ ----------
Mon Dec 29 21:26:04 2008 DF-7A  Symm Communication Warning      0x001a
    The Symmetrix Service Processor could not complete a Call Home for service
```

List The following example illustrates how to retrieve a verbose list of the events which have occurred on the specified Symmetrix array between 12 p.m. and 3 p.m. today:

```
symevent list -sid 0207 -start 12:00 -end 15:00 -v

Symmetrix ID: 000192600207
Time Zone   : EST

Event at Mon Dec 29 13:00:10 2008:
   Reporting Director : RF-7H
   Source             : Symmetrix
   Category           : RDF
   Severity           : Informational
   Numeric Code       : 0x000d
   Event Code Symbol  : RDF_SIM_MESSAGE
   Description        : The RDF subsystem initiated a SIM message to a Symmetrix
 remotely-attached via RDF links

Event at Mon Dec 29 13:01:38 2008:
   Reporting Director : FA-7E
   Source             : Symmetrix
   Category           : RDF
   Severity           : Informational
   Numeric Code       : 0x0018
   Event Code Symbol  : RDF2_DEVICE_NR
   Description        : One of the RDF2 devices was found to be Not Ready
```

Common audit log

Data is written to a common audit file during Symmetrix control operations initiated by host applications. The common audit log correlates activity from all hosts into one file that is stored in the Symmetrix File System (SFS).

The `symaudit` command enables the examination and filtering of the common audit log file for a specified Symmetrix array. The audit log resides on the Symmetrix array and currently has a maximum size of 40 MB. Once the 40 MB limit is reached, the log begins to overwrite itself. You do not 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 or monitor 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, hostname, username, and device acted upon.
- `monitor` — Monitors the Symmetrix array for new audit log data in realtime.

Note: For details on valid parameter values on which to filter, refer to the `symaudit` command in the *Solutions Enabler SYMCLI Command Reference HTML Help*.

Example output

The following is an example output of the `show` action for a specified Symmetrix array:

```
symaudit show -sid 0207

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

Symmetrix ID             : 000192600207

Starting date            : 12/17/2008 21:42:57
Ending date              : 01/02/2009 15:30:35

Starting record number   :       1
Ending record number     :    1737
Total record count       :    1737
```

The following is an example output of the `list` option for a specified Symmetrix array between a specified time period:

```
symaudit list -sid 0207 -start_time 01/02/2009:12:00:00 -end_time
01/02/2009:12:15:00 -v

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

Symmetrix ID             : 000192600207

  Record Number          :     1663
    Records in Seq       :        2
    Offset in Seq        :        1
    Time                 : 01/02/09 12:11:50
    Vendor ID            : EMC Corp
    Application ID       : SYMCONFIGURE
    Application Version  : 7.0.0.302
    API Library          : SDK
    API Version          : T7.0.302.2 (Edit Level: 907)
```

```
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : STARTING a Device Reservation 'RELEASE'. Owner=m; Rese
rveID=1; Comment="t";
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

  Record Number         :     1664
    Records in Seq      :        2
    Offset in Seq       :        2
    Time                : 01/02/09 12:11:50
    Vendor ID           : EMC Corp
    Application ID      : SYMCONFIGURE
    Application Version : 7.0.0.302
    API Library         : SDK
    API Version         : T7.0.302.2 (Edit Level: 907)
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : Devices: [ 0020 ]
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

  Record Number         :     1665
    Records in Seq      :        1
    Offset in Seq       :        1
    Time                : 01/02/09 12:11:51
    Vendor ID           : EMC Corp
    Application ID      : SYMCONFIGURE
    Application Version : 7.0.0.302
    API Library         : SDK
    API Version         : T7.0.302.2 (Edit Level: 907)
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : Device Reservation 'RELEASE' SUCCEEDED. ReserveID=1;
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

<. . .>
```

- Record Number — .The current record number.
- Records in Seq — Total number of records requested.
- Offset in Seq — Offset number from the first record requested.
- Time — Date and time the record was entered.
- Vendor ID — ID of the vendor whose application logged the record.
- Application ID — ID of the application that logged the record.

- Application Version — Application version number.
- API Library — Name of the SYMAPI library the application ran against.
- API Version — Version of the SYMAPI.
- Host Name — Name of the host that logged the record.
- OS Name — Operating system on which the host is running.
- OS Revision — Operating system revision number.
- Client Host — Any SYMCLI client communicating with the SYMAPI server.
- Process ID — ID of the process that logged the record.
- Task ID — ID of the task.
- Function Class — Class name of the SYMAPI functional area.
- Action Code — Name of the SYMAPI control action associated with an audit log entry.
- Text — Text details of the given entry.
- Username — Identifies the user that generated the log entry.
- Activity ID — A randomly generated ID that uniquely identifies this action.

Event daemon

Solutions Enabler also provides an asynchronous event daemon that can be used to monitor and manage Symmetrix events. The event daemon (`storevntd`) provides the necessary services required to monitor the status of Symmetrix storage environments from third-party enterprise management frameworks. The following targets are supported:

- SNMP
- file on disk
- system logger on the host
- Unix syslog service
- Windows event log
- syslog listener across the network (bypasses the syslog service (calls) on the local host and directly sends events/traps to this remote listener).

To learn more about enabling and configuring the event daemon, refer to the *EMC Solutions Enabler Installation Guide*.

12

XML Structured Output

This chapter describes the XML output option of the SYMCLI.

XML structured output data

The XML (Extensible Markup Language) output option provides a mechanism to facilitate the automated processing of SYMCLI output data. XML is a deterministic parsing tool that eases parsing of output data, providing a functional advantage over screen-scraping tools like awk or Perl. The XML industry standard is based on the experience of SGML and is endorsed by the World Wide Web Consortium. Detailed information on XML may be found at:

```
http://w3.org/XML/
```

XML has the look and feel of HTML, as it employs the same tag-based syntax. However, XML uses tags to delimit data—as opposed to defining the data as with HTML—allowing the document author to specify the tags most applicable to the given application. For the SYMCLI, tags represent the physical and logical structures within the Symmetrix array and its environments.

When XML mode is utilized, the data returned is identical to that of the *standard* output, but "marked-up" with tags. These tags enable individual pieces of data to be readily called upon by name. In addition, they provide a definitive way to express the relationship between different objects, an advantage over the standard CLI display output.

XSLT: XML data transformations

Many tools are available to query, filter, retrieve, and format specific information stored in complex XML files. Among these, eXtensible Stylesheet Language Transforms (XSLT) is a particularly useful and widely available technology. While using XML will result in less ambiguous, more robust scripts, XSLT will make the information presented in XML accessible to the more familiar plain text-based scripting techniques. To introduce you to XSLT, a directory containing several examples of the types of queries that can be performed on XML data using XSLT is provided. The examples are designed only to provide a brief introduction to the power and usefulness of XSLT, and can help ease the transition to XML.

XML output options

In XML, data can be described either by using elements or attributes, or the combination of both. Versions of Solutions Enabler prior to version 6.3 only provided attribute-based XML data. Beginning in version 6.3, you can return element-based XML output.

Element-based XML describes data in a hierarchical manner by using the notion of parent and children. An element can have several different content types. It can have element content (child element), a mixed content which contains both text and child element, a simple content containing text only, or an empty content carrying no information. An element can also have attributes. These additional content types would allow users to modify the data structures in a fairly flexible manner. On the other hand, an attribute is used to provide additional information about an element. An attribute is in general used to store the metadata describing the data that stored in XML. Although data can be stored in attributes, it is best practice to store data in child elements.

Note: The attribute-based XML output has been deprecated beginning with Solutions Enabler version 6.3 features and is maintained strictly for backwards compatibility. To view new features, use the element-based XML output (`xml` or `xml_element`).

Using XML Mode

To use XML mode with SYMCLI, an environment variable or a command line option can be used. Use the environment variable to globally set your command output to XML or globally return it to standard mode. Use the following syntax to set the environment variable:

```
SYMCLI_OUTPUT_MODE = xml|xml_element|xml_attribute|standard
```

- **xml** — Returns the output of all legacy commands in element-based XML tags.

  Note: In versions of Solutions Enabler prior to V6.5, the `xml` value returns attribute-based XML.

- **xml_element** — Returns the output of all commands in element-based XML tags.
- **xml_attribute** — Returns the output of all commands in attribute-based XML tags.
- **standard** — Returns the output of all commands to the default output without the XML tags.

Note: When the environment variable output mode is set to `xml`, `xml_element`, or `xml_attribute`, 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`. This will allow you to successfully execute the given command in standard mode.

The command line method provides a means of outputting any single command as either XML or standard output. Using the command line method overrides the current environment variable setting:

```
<SymcliCommand> -output <xml|xml_element|xml_attribute|standard>
```

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

Example Output

The following examples show some of the differences between the element- and attribute-based XML output.

Element-based XML

Notice that a new element tag `<Symm_Info>` has been added to the XML data shown below to store the general Symmetrix data.

`symcfg list -out xml` OR `symcfg list -out xml_element`

```
<?xml version="1.0" standalone="yes" ?>

<SymCLI_ML>
  <Symmetrix>
    <Symm_Info>
      <symid>000190102055</symid>
      <attachment>Local</attachment>
      <model>DMX3-24</model>
      <microcode_version>5772</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <devices>683</devices>
      <physical_devices>94</physical_devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000190300215</symid>
```

```
        <attachment>Local</attachment>
        <model>DMX3-6</model>
        <microcode_version>5772</microcode_version>
        <cache_megabytes>32768</cache_megabytes>
        <devices>239</devices>
        <physical_devices>2</physical_devices>
      </Symm_Info>
    </Symmetrix>
    <Symmetrix>
      <Symm_Info>
        <symid>000190300237</symid>
        <attachment>Remote</attachment>
        <model>DMX3-6</model>
        <microcode_version>5773</microcode_version>
        <cache_megabytes>16384</cache_megabytes>
        <devices>3756</devices>
        <physical_devices>0</physical_devices>
      </Symm_Info>
    </Symmetrix>
    <Symmetrix>
      <Symm_Info>
        <symid>000190300343</symid>
        <attachment>Remote</attachment>
        <model>DMX3-6</model>
        <microcode_version>5773</microcode_version>
        <cache_megabytes>32768</cache_megabytes>
        <devices>751</devices>
        <physical_devices>0</physical_devices>
      </Symm_Info>
    </Symmetrix>
</SymCLI_ML>
```

Note: To maintain consistent element names in all SYMCLI commands, some tag names will be redefined. For example, in current SYMCLI command, different names exist to describe Symmetrix identification number, such as id, symmetrix, or symid. A new consistent tag name will be defined across all SYMCLI commands in the element-based XML output.

Attribute-based XML

```
symcfg list -out xml_attribute

<?xml version="1.0" standalone="yes" ?>

<SymCLI_ML>
  <Symmetrix id="000190102055" attachment="Local" model="DMX3-24"
    microcode_version="5772" cache_megabytes="32768"
    physical_devices="94" devices="683"/>
  <Symmetrix id="000190300215" attachment="Local" model="DMX3-6"
microcode_version="5772" cache_megabytes="32768"
physical_devices="2" devices="239"/>
  <Symmetrix id="000190300237" attachment="Remote" model="DMX3-6"
    microcode_version="5773" cache_megabytes="16384"
    physical_devices="0" devices="3756"/>
  <Symmetrix id="000190300343" attachment="Remote" model="DMX3-6"
    microcode_version="5773" cache_megabytes="32768"
    physical_devices="0" devices="751"/>
</SymCLI_ML>
```

Index

T

U

V

X