

**EMC Solutions Enabler
Symmetrix Optimization CLI**
Version 6.0

PRODUCT GUIDE

**P/N 300-000-878
REV A06**

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

Copyright © 2002, 2003, 2004, 2005 EMC Corporation. All Rights Reserved.

Printed January, 2005

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

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

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

Trademark Information

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

Preface	vii
----------------------	-----

Chapter 1 Introduction

Introduction to Performance Optimization	1-2
Symmetrix Optimizer.....	1-2
Quality of Service.....	1-3

Chapter 2 Symmetrix Optimizer

What Is Symmetrix Optimizer?	2-2
How Optimizer Works.....	2-2
Supported Environments	2-3
Configuration Requirements.....	2-4
Dynamic Reallocation Volumes (DRVs)	2-4
Interaction with TimeFinder	2-4
Open Mirror Position Requirement	2-4
Open Configuration Lock Requirement.....	2-5
How Does Symmetrix Optimizer Work?.....	2-6
The Analysis	2-6
Performance Metrics	2-7
Modeling Service Time	2-7
Finding the Best Swap.....	2-8
The Swap Procedure.....	2-9
Legality of Swaps.....	2-11
Impact of Swaps.....	2-13
Symmetrix Optimizer Operations via SYMCLI	2-14
Starting and Stoppping Optimizer.....	2-14
Committing Command File Changes	2-14
Setting User Approved Mode.....	2-15

	Setting Performance Time Windows	2-15
	Setting Swap Priority	2-17
	Performing Rollbacks.....	2-18
	Optimizer Logs and Swap History	2-18
	Command File General Parameters	2-19
	Example	2-20
Chapter 3	Quality of Service	
	Overview.....	3-2
	Setting QoS Copy Parameters and Values	3-3
	Copy Pace to Certain Devices.....	3-3
	Setting QoS LRU Parameters and Values.....	3-5
	LRU Cache I/O Management.....	3-5
	Viewing QoS Parameters	3-6
Index		i-1

Tables

1-1	Symmetrix Optimizer Command	1-3
1-2	SYMAPI Quality of Service Command	1-3
2-1	Optimizer Strategies for Determining Swaps	2-6
2-2	Parameters in the Command File	2-19

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

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

Audience This manual is intended for command-line users and script programmers interested in details about using the SYMCLI commands to optimize the Symmetrix environment.

Organization The manual describes how to optimize and execute certain control operations on your storage environment using the SYMCLI commands of the EMC Solutions Enabler software. The following defines the structure of this manual:

Chapter 1, *Introduction*, introduces the various performance optimizations that can be made to your Symmetrix environment.

Chapter 2, *Symmetrix Optimizer*, describes how Symmetrix Optimizer works and how to manage the Symmetrix Optimizer using SYMCLI.

Chapter 3, *Quality of Service*, shows you how to view and control quality of service device parameters.

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

Related Documentation

Related documents include:

- ◆ *EMC Solutions Enabler Symmetrix CLI Command Reference*, EMC Corporation
- ◆ *MC Solutions Enabler Installation Guide*, EMC Corporation
- ◆ *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*, EMC Corporation
- ◆ *EMC Symmetrix Optimizer Engineering White Paper*
- ◆ *EMC Solutions Enabler Version 6.0 Support Matrix*
- ◆ *Symmetrix High Availability Product Guide*

Conventions Used in this Manual

The following conventions are used in this manual:

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

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



CAUTION

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

Typographical Conventions

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

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

Where to Get Help

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

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

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

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

Sales and Customer
Service Contacts

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

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

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

<http://powerlink.emc.com>

Your Comments

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

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

Introduction

This chapter provides an introduction to the various performance optimizations that can be made to your Symmetrix environment. It provides a brief overview of Symmetrix Optimizer and QoS.

- ◆ Introduction to Performance Optimization1-2

Introduction to Performance Optimization

The SYMCLI optimization commands provide performance features and controls beyond those provided by the base component. The SYMCLI control commands are grouped in the following categories, and described in the upcoming pages:

- ◆ Symmetrix Optimizer
- ◆ Quality of Service (QoS)

For a detailed introduction to Solutions Enabler, SYMCLI, and the Symmetrix array, refer to the *EMC Solutions Enabler Symmetrix Base Management CLI Product Guide*. Detailed manpage descriptions of the optimization commands described below can be found in the companion *EMC Solutions Enabler Symmetrix CLI Command Reference*.

Symmetrix Optimizer

The Symmetrix Optimizer is a feature that performs self-tuning of Symmetrix storage configurations by collecting and analyzing statistics about Symmetrix device activity. It then determines which Symmetrix device hyper-volume locations could be swapped to produce the optimum Symmetrix back-end performance. It then swaps Symmetrix device hyper-volume locations using internal Dynamic Reallocation Volumes (DRVs) to hold storage enterprise data while hyper reconfiguration executes.

The version of Symmetrix Optimizer that is installed on a Symmetrix depends on the version of Enginuity™ (microcode) and the version of SYMCLI. The `symoptmz` command sets parameters that control the behavior of the Symmetrix Optimizer algorithm.

The SYMCLI `symoptmz` command supports Symmetrix Optimizer version 2 and above.

Table 1-1 describes the main features of the Symmetrix Optimizer command:

Table 1-1 Symmetrix Optimizer Command

Command	Description
symoptmz	Performs operations on the Symmetrix Optimizer. Possible operations are: • Lists information about the Optimizer-specific device attributes of Symmetrix devices that are configured in one or more Symmetrix units connected to this host. • Shows specific information about the current Optimizer parameters. • Enables the Optimizer algorithm processing. • Disables the Optimizer algorithm processing. • Queries Symmetrix Optimizer and displays the current state and version information. • Clears disk statistics maintained by the Optimizer. • Controls the Optimizer session and releases the Optimizer's API lock. • Rolls back the Symmetrix configuration to that of a previous point-in-time. • Verifies, performs range checks, and commits a command file.

For more information on the `symoptmz` command, refer to Chapter 2.

Quality of Service

The `symqos` command provides Quality of Service controls on specified devices. Table 1-2 lists the features of the QoS command:

Table 1-2 SYMAPI Quality of Service Command

Command	Description
symqos	Invokes quality of service controls on certain devices. Possible operations are: • Displays a list of quality of service priorities assigned to devices and mirrors. • Sets the values and parameters for QoS priorities on the devices and mirrors. • Queries the QoS copy priorities for members of a specified group. • Lists the LRU cache assignments. • Sets the LRU cache assignments to a device or device group. • Resets the LRU assignment back to the default LRU.

For more information on the `symqos` command, refer to Chapter 3.

Symmetrix Optimizer

This chapter describes how to adjust the parameters of the Symmetrix Optimizer with SYMCLI.

- ◆ What Is Symmetrix Optimizer?2-2
- ◆ Configuration Requirements.....2-4
- ◆ How Does Symmetrix Optimizer Work?.....2-6
- ◆ Finding the Best Swap2-8
- ◆ Symmetrix Optimizer Operations via SYMCLI.....2-14

What Is Symmetrix Optimizer?

Symmetrix Optimizer is one of a family of information management software solutions offered by EMC. Symmetrix Optimizer improves array performance by continuously monitoring access patterns and migrating devices (Symmetrix logical volumes) to achieve balance across the disks in the array. This automated process is based on user-defined parameters and is completely transparent to end users, hosts, and applications in the environment. Migration is performed with constant data availability and consistent protection.

This chapter describes how to manage Symmetrix Optimizer via the the SYMCLI. Using the Symmetrix Optimizer SYMCLI command, `symoptmz`, users can define:

- ◆ Symmetrix devices to be optimized
- ◆ Priority of those devices
- ◆ Window of time that profiles the business workload
- ◆ Window of time in which Optimizer is allowed to swap
- ◆ Additional business rules
- ◆ The pace of the Optimizer volume copy mechanism

How Optimizer Works

Once initialized with the user-defined parameters, Symmetrix Optimizer operates totally autonomously on the Symmetrix service processor. Symmetrix Optimizer monitors and controls array performance via the following steps:

Step 1: Symmetrix Optimizer builds a database of device activity statistics on the Symmetrix back-end.

Step 2: Using the statistical data collected, configuration information, and the user-defined parameters, the Optimizer algorithm identifies busy and idle devices and their locations on the physical drives. The algorithm tries to minimize average disk service time by balancing I/O activity across physical disks. Optimizer determines which disks require balancing by locating busy devices close to each other on the same disk, and/or by locating busy devices on faster disks or faster areas of the disks. Optimizer takes into account the speed of the disk, the disk geometry, and the actuator speed to determine faster disks.

Step 3: Once a solution for load balancing has been developed, the next phase is to carry out the Symmetrix device swaps. This is done

using established EMC® TimeFinder® technology, which maintains data protection and availability. Users can specify if swaps should occur in a completely automated fashion, or if the device swaps require user approval before the action is taken.

Step 4: Once a swap function completes, Symmetrix Optimizer continues data analysis for the next swap.

Supported Environments

Symmetrix Optimizer consists of two major components: Optimizer processing engine and a client. The Optimizer processing *engine*, which runs continuously on the Symmetrix service processor and collects the statistics, performs the analysis, and executes the migrations. This component is associated with the version of Symmetrix Enginuity software running on the Symmetrix array. Versions of the Optimizer engine (with different functionality) are referred to as different *revs* of Optimizer.

Note: Refer to the *EMC Solutions Enabler Support Matrix* and the *EMC ControlCenter Support Matrix* (found on the Powerlink website) for the required minimum versions of Enginuity software release levels, along with the compatible user control interfaces for each revision of Optimizer and the required service processor versions.

Since Symmetrix Optimizer operates on Symmetrix devices and physical disks, there are no compatibility requirements at the host or application level.

Symmetrix Optimizer can optimize all open system and mainframe RAID 1 (mirrored) and RDF-protected volumes in an array. Optimizer does not work with volumes configured for use with the AS/400 operating system. Optimizer does not consider business continuance volumes (BCVs) or dynamic reallocation volumes (DRVs) for swapping (refer to *Legality of Swaps* on page 2-11 for a complete list of valid swaps).

Note: As of September 2003, Optimizer does not support parity RAID (parity 3+1, 7+1) and RAID 1/0 volumes.

Configuration Requirements

Symmetrix Optimizer must be enabled by EMC from the service processor and DRV's must be configured on the array. In addition, Optimizer requires of an open mirror position to perform it's swapping function. If users are running Enginuity 5x66 or 5x67, it is critical to be aware of Optimizer's potential interaction with TimeFinder activity.

Dynamic Reallocation Volumes (DRVs)

The DRV is a non-user-addressable logical volume used by Symmetrix Optimizer to temporarily hold customer data while executing reconfiguration (on logical volume granularity). Data is available and protected during the reconfiguration process.

As a minimum, two DRVs must be configured for each size and emulation of the volume to be swapped by Optimizer. For example, for Optimizer to swap a 4 GB open-system volume, then two 4 GB open system DRVs must be configured. If more than one simultaneous swap is desired, then additional DRVs are needed (two for each swap). If there are volumes of mixed size or emulation within the same array, then DRVs are required for each size and type to be swapped.

Interaction with TimeFinder

With the introduction of the features for concurrent BCVs and independent splits in Enginuity Version 5568.49.18, the direct conflicts between TimeFinder and Optimizer have been eliminated.

Open Mirror Position Requirement

Optimizer requires an open mirror position to be able to perform a swap, however, Symmetrix arrays have a four mirror slot limitation per volume. This limitation may make it difficult in certain configurations for Optimizer to obtain the mirror position it requires to perform a swap. For example, a RAID 1 device already has two BCVs established, and Optimizer wants to swap this device, it cannot get a mirror slot for the DRV. If Optimizer does not have a mirror slot available, then it will drop the swap and rerun the analysis. If Optimizer is in Rollback, Manual, or User-Approved mode, then Optimizer will retry a number of times until an error is encountered. Eventually, Optimizer will stop trying if a mirror slot does not become available.

Open Configuration Lock Requirement

Optimizer also must hold a Symmetrix External Lock (SEL) to perform a swap. The configuration lock is an exclusive lock on the Symmetrix array for performing configuration changes. This lock prevents multiple applications from changing the array configuration at the same time. If another application (such as SDR) holds the configuration lock when Optimizer wants to create a swap, then Optimizer will behave as previously described (no open mirror position). Optimizer will hold the configuration lock for the duration of the swap, so other applications will not have the ability to perform configuration changes during this time.

How Does Symmetrix Optimizer Work?

The analysis methodology and the swap procedure are the core technology and the two main components of the Symmetrix Optimizer solution. The following sections provide a brief description of each.

The Analysis

When Symmetrix Optimizer evaluates the performance statistics, it determines potential volume swaps based on how well they would improve overall performance. This analysis is based on minimizing disk service time (rotational latency plus seek time plus transfer time). Symmetrix Optimizer uses three strategies when determining which swaps to make:

Table 2-1 Optimizer Strategies for Determining Swaps

Swap Strategy	Description
1. Load Balancing 	Swaps highly active volumes to disks with lower activity evens out the load across the physical drives in the Symmetrix system. This decreases the contention on the individual physical disks, improving the performance of both highly active volumes and low-activity volumes.
2. Minimize Seek 	Performs swaps that relocate highly active volumes so that they are closer together on the physical disk and will decrease the seek distance for I/Os to these volumes. When Optimizer swaps volumes, it will try to achieve this type of configuration since it decreases the overall I/O service time.
3. Use Faster Media 	Optimizer tries to swap highly active volumes on the outer zones of the disk. This is because volumes located on the outer zones of the disk have faster transfer speeds.

Symmetrix Optimizer's algorithm uses detailed disk performance information that takes into account several drive characteristics, such as those gig-to-gig seek times, zone-bit recording, and bandwidth data.

Performance Metrics

Optimizer looks only at back-end activity; it uses the back-end logical volume statistics, and the metrics used are:

- ◆ DA logical volumes reads
- ◆ DA logical volumes writes
- ◆ Logical volume prefetch
- ◆ DA logical volumes Kb read
- ◆ DA logical volumes Kb written

While modeling the disk service time, different weights are assigned to read, write, and prefetch activity. The assumption is that since writes are done as a background process, they are done in sequence and hence are more efficient. Optimizer does not try to follow DMSP policies; instead, it assumes that reads are equally spread among all the device mirrors, and all mirrors perform all writes.

Modeling Service Time

Service time is defined as the sum of seeks, latency and transfer time.

- ◆ **Seek time** — Seek time is the time it takes the disk arm to move and position the disk head on the correct track — move from track X (serving the previous I/O) to track Y (serving the next I/O). Optimizer uses a gig-to-gig database to model seek time for different addresses and different disk drives.
- ◆ **Latency time** — Latency time is the delay for disk rotation. The latency time is a function of the disk rotational speed. Optimizer assumes one-third of a spin for each I/O, assuming the internal disk optimization is on.
- ◆ **Transfer time** — Transfer time is the time that it takes the disk to transfer the data from/to the disk. Transfer time is a function of the data transfer rate, disk bandwidth, and data layout. Optimizer uses the Zone Bit Recording (ZBR) database to model the transfer time of data. The ZBR database includes information about the bandwidth of each zone of the disk.

Accurate seek and latency times are impractical to get because they require a complete trace of I/Os sequence, therefore the Optimizer uses a mathematical model to calculate these metrics.

Finding the Best Swap

The Optimizer analysis consists of three high-level phases:

- ◆ **Calculate service time:** Model and sum the total service time of each disk and for every time stamp that was marked to be included by the analysis.
- ◆ **Sort disks by activity:** Sort all disks by their modeled total service time.
- ◆ **Find best swap:** Starting at the busiest disk, check all potential swaps. The analysis process models *what-if* scenarios using *virtual swaps* to estimate the impact of a swap on the service time of the affected disks. The philosophy of Optimizer is to check as many swaps as possible in order to guarantee that the best swap is indeed selected.

Currently, EMC allows 256 hypers per disk; finding a swap that improves service time by 10 percent or more on these systems is very rare. Since using a percentage of improvement as swap *goodness* criteria is not applicable anymore, Optimizer had to adapt a different method (*M0*) that deals better with the “many hypers contribute smaller chunks” problem. *M0* is defined as “the best you could possibly get” from a swap. Each disk in the system is assigned an *M0* designation, which is defined as the minimum service time a disk can get by replacing one of its hypers by a null hyper (a hyper that performs no I/Os to the disk).

When analyzing a swap, Optimizer checks how close the new-modeled service time is to the disk’s *M0*; the closer to *M0*, the better the swap. Usually, two disks are affected by a single swap; the busier disk’s service time is expected to go down, while the other disk’s service time is expected to go up. In addition to the *M0*, the Optimizer also ensures that the new maximum of service times is less than the old one.

The Swap Procedure

The swap procedure relies on the Symmetrix TimeFinder technology and uses DRVs as temporary mirrors. The following section describes the four swap steps when dealing with mirrored devices swaps. In the case of three mirror devices, the swap procedure does not require a DRV.

Hypervolumes are swapped using a four-step process. To swap two hypervolumes, the hypervolumes must be the same size, the same emulation, and, based on the architecture, on the same system bus or in the same power zone in the Symmetrix DMX™ architecture. At least two DRVs must be configured per swap. With Version 5.1.1 of Symmetrix Optimizer, up to eight simultaneous swaps can happen at one time.

Step 1: Identify Volumes to Swap

Symmetrix Optimizer identifies a pair of hypervolumes to swap by recognizable patterns of hypervolume activity and criteria. Assume the red volumes (on A) have high activity and the blue volumes (on B) have low activity.

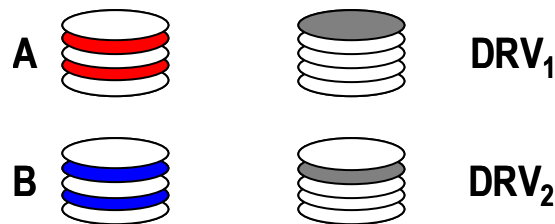


Figure 2-1 Identify Volumes to Swap

Step 2: Copy Volume to DRV

Symmetrix Optimizer swap commands are passed to SymmWin, which assigns one DRV (DRV1) as a third mirror for hypervolume A. A second DRV (DRV2) will be assigned as a third mirror for hypervolume B. All tracks on the third mirror are marked invalid. Tracks are copied from the valid mirrors to the two DRVs. After the DRVs are synchronized, the two original swap physical mirrors are marked Not Ready (volume A and volume B) and their attributes are swapped (see Figure 2-2). Both hypervolumes still have two (or more) physical mirrors. Host activity to the hypervolumes is now directed to DRVs and the other mirror.

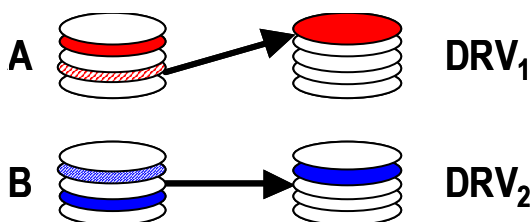


Figure 2-2 Copy Volume to DRV

Step 3: Copy DRV to New Location

The data on the DRVs are now copied to the new location. After the attributes of the original hypervolumes are swapped, SymmWin copies the tracks from the valid mirrors to the two new mirrors and then makes the original hypervolumes Ready (see Figure 2-3). This is similar to a BCV restore.

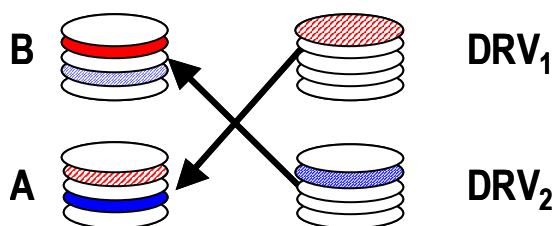


Figure 2-3 Copy DRVs to New Locations

Step 4: DRVs are split

The final step is to split the DRVs from their standard and mirror hypervolumes after synchronization completes (see Figure 2-4). The drive's balance improves and the DRVs are now available for the next swap.

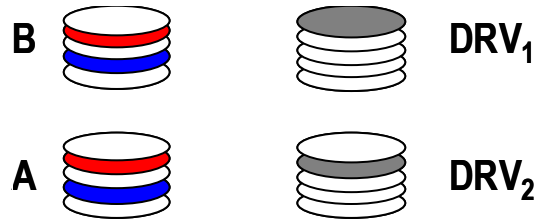


Figure 2-4 DRVs Are Split

Note: Swap of three mirror devices does not require a DRV.

Legality of Swaps

As a rule, Optimizer will never suggest a swap that conflicts with configuration rules as defined by the Enginuity, SymmWin, or the configuration groups. For Symmetrix 3000, 4000, and 8000 hardware families, all configuration rules are static. However, for the Symmetrix DMX and RMS families, some configuration rules are static and some are more dynamic and defined in relative manner where an Optimizer swap can improve or maintain protection level but will never decrease protection level. (Exceptions for this might be a rollback session, where the user wants to undo a swap.)

Enginuity 67/68 (Symmetrix 3000, 4000, and 8000 Families)

For Symmetrix 3000, 4000, and 8000 hardware families, the following rules apply:

- ♦ Swappable devices
 - RAID 1 (mirrored volumes)
 - RDF devices with two local copies (default, can be controlled from SP)
 - Same size and emulation
- ♦ Not swappable devices – The following devices are not swappable:
 - Parity RAID (parity and data)
 - CKD Stripe (RAID 1/0)
 - BCV, DRV

- SFS
- WORM
- AS400
- Standard, unprotected volumes (default, can be controlled from SP)
- ◆ Other configuration restrictions:
 - **Same disk** — Two mirrors of the same device should never reside on the same disk (for example, 1a:C0).
 - **X/Y bus** — Two mirrors of the same RAID 1 device should never reside on the same memory bus (for example, 1a with 3a).
 - **Dual-Initiator** — Two mirrors of the same RAID 1 device should never reside on the same DA and dual-initiator pair (for example, 1a with 2a).

Enginuity 69/70 (Symmetrix DMX and RMS)

Configuration rules for the Symmetrix DMX and RMS families are relative in manner. Optimizer can improve or maintain protection, but never decrease level of protection. However, some restrictions should always be followed.

- ◆ Swappable devices
 - RAID 1 (mirrored volumes)
 - RDF devices with two local copies (default, can be controlled from SP)
 - Same size and emulation
- ◆ Not swappable devices — The following devices are not swappable:
 - Parity RAID (parity and data)
 - CKD Stripe (RAID 1/0)
 - BCV, DRV
 - SFS
 - WORM
 - AS400
 - SAVE devices and VDEVs
 - Standard, unprotected volumes (default, can be controlled from the Service Processor).

- Targets of Snap sessions
- ◆ Other configuration restrictions:
 - **Same disk** — Two mirrors of the same device should never reside on the same disk (for example, 1a:C0).
 - **Same port** — Two mirrors of the same device should never reside on the same SCSI port (for example, 1a:C).
 - **Same loop** — All mirrors of the same RAID 1 device should never reside on the same disk loop (for example, 1a:C with 16a:C).

Note: Not applicable to RMS.

- **Power zones** — All mirrors of the same RAID 1 device should never reside on the same power zone. [Applicable to Panther and Rhino only]
- **Same DA** — All mirrors of the same RAID 1 device should not reside on the same DA (CPU) (for example, 1a).
- **Same slot** — All mirrors of the same RAID 1 device should not reside on the same director (slot) (for example, 1a with 1b).
- **Dual-Initiator** — All mirrors of the same RAID 1 device should not reside on the same dual-initiator pair (for example, 1a with 16a).

Note: Not applicable to RMS.

Impact of Swaps

Each swap performed by Optimizer is the equivalent of two TimeFinder operations. To minimize the affect of the swap procedure on the overall Symmetrix performance, the Optimizer will not include the same spindle twice in one swap group. In addition, the customer, using QoS controls, can set a lower BCV priority for the STDs to be swapped, and can plan the swap activities to occur on idle business time using the Time Window feature.

Symmetrix Optimizer Operations via SYMCLI

The Symmetrix Optimizer is a component of the EMC ControlCenter™ Symmetrix Manager. The SYMCLI on the host machine issues commands to display and set the values of the parameters that control the behavior of the Symmetrix Optimizer.

The Symmetrix Optimizer SYMCLI commands enables users to define:

- ◆ Symmetrix devices to be optimized
- ◆ Priority of those devices
- ◆ Window of time that profiles the business workload
- ◆ Window of time in which Optimizer is allowed to swap
- ◆ Additional business rules
- ◆ The pace of the Symmetrix Optimizer volume copy mechanism

Starting and Stopping Optimizer

To start and stop the Optimizer process on the array service processor, use the `symoptmz enable` and `symoptmz disable` commands. Once disabled, the Optimizer process on the Symmetrix service processor will still listen for Optimizer clients (ControlCenter or SYMCLI), but it will not collect statistics or initiate any swaps until it is enabled.

Committing Command File Changes

To change any Optimizer parameters, Optimizer must be stopped using the `symoptmz disable` command. The parameters are then set using a command file (for command file parameters, refer to *Command File General Parameters* on page 2-19); then, validated and committed using the `preview`, `prepare`, and `commit` options. For example, to commit the changes specified in the command file `opt_config.txt`, issue the command:

```
symoptmz -file opt_config.txt commit
```

The `preview` argument checks that the command file has correct syntax. The `prepare` argument also checks syntax and, in addition, performs some range checks. The `commit` argument carries out the same syntax and range checks, and then updates the Optimizer with the modified parameters. It is recommended that a `preview` be run on your command files, and then any syntax errors corrected. A `prepare` should then be run and any out-of-range figures should be corrected.

Finally, a `commit` should be done. The `commit` command will first do a preview and prepare before committing the settings in the command file.

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.

Setting User Approved Mode

User Approved mode allows you to see what swaps will occur before they take place. To switch Optimizer to User Approved mode, use the `set control_parms swap_mode=USER_OK;` line in the command file. Once this is set, Optimizer recalculates a swap list approximately once every hour as long as samples have been collected for the specified minimum performance period. At any time, the latest swap list can be retrieved using the `symoptmz show -swap_list` command. The swap list can then be approved or declined using the following syntax in the command file:

```
set swap <APPROVE | DECLINE>
[begin_at=<time_val>,)
TIMESTAMP=<time_val>;
```

The time stamp specified must be the time stamp returned by the last `symoptmz show swap_list` command. If the command to approve or decline the swap list returns an error, the swap list is probably out of date and a new one is available from the service processor. The latest swap list should then be retrieved again with the `symoptmz show -swap_list` command.

Setting Performance Time Windows

To set the performance time windows, use the `symoptmz` command with the following syntax in the command file:

```
set time_window id=<tw_id>,
type=<SWAP | PERF>,
flag=<INCLUDE | EXCLUDE>,
period=<ONCE | WEEKLY | WEEKLY_BY_DAY>,
starting=<date_time>,
ending=<date_time>,
[days=<day_list>,
start_time=<hh:mm>, end_time=<hh:mm>;
```

where:

- ◆ `<tw_id>` is a user-defined name that identifies this time window.

- ◆ `type` defines the kind of time window being defined. Two types are available:
 - `SWAP`: Defines whether a swap is (`INCLUDE`) or is not (`EXCLUDE`) performed in the given time window, according to the `flag` setting.
 - `PERF`: Defines whether a monitoring is (`INCLUDE`) or is not (`EXCLUDE`) performed in the given time window, according to the `flag` setting.

Using this setting, you could, for example, configure Optimizer to monitor on weekdays from 9 - 5 and only allow swaps on Sundays.

- ◆ `flag` defines the option for the time window type:
 - `INCLUDE`
 - `EXCLUDE`
- ◆ `period` defines the frequency of occurrence:
 - `ONCE` means the action will occur only one time.
 - `WEEKLY_BY_DAY` requires the individual days of the week to be specified. The time window applies to each day specified and each day has its own time window.
 - `WEEKLY` setting is equivalent to a *range*. The `starting` parameter identifies the first of a series of consecutive days that the recurring time period applies to, and the next day in the list serves as the ending day of the week to which the time period applies.
- ◆ `<date_time>` is in the form of `MMDDYYYY:HHMMSS`.
- ◆ `<day_list>` is any comma-separated combination of `MON`, `TUE`, `WED`, `THU`, `FRI`, `SAT`, or `SUN`. For the case of `WEEKLY`, `<day_list>` must begin with one of the following: `MON_START`, `TUE_START`, `WED_START`, `THU_START`, `FRI_START`, `SAT_START`, or `SUN_START`, which identifies the first of a series of consecutive days to which the time window applies. The next entry identifies the day of the week, which concludes the range of days.

Using the `set time_window` command in the command file erases any previously set time windows (both swap and performance), so this command should be used with caution.

To list the performance time windows that have been set up, use the `symoptmz show -composite` command. The `symoptmz show -parms` command will list any time windows that have been set.

Setting Swap Priority

Use the following command to set the volume priority:

```
symoptmz set swap_priority <none|normal|high>  
[-range [Startdevname]:[Enddevname]] [-n NumDevs]
```

Use the following command to list the priority settings for all the volumes or a range of volumes:

```
symoptmz list [-range [Startdevname]:[Enddevname]]  
[-n NumDevs]
```

Performing Rollbacks

Rollbacks are used to undo swaps that conflict with your business rules. Rollbacks are performed by first placing the Symmetrix Optimizer in rollback mode by calling `symoptmz rollback -sid SymmID` on a specified Symmetrix. To return a list of possible rollback points use the `symoptmz -rollback_list -sid SymmID` command.

The rollback feature is an *all or nothing* feature. Optimizer reverses all swaps by going backward from the present to a selected earlier time and undoing each and every swap. Use the User Approved mode as described in *Setting User Approved Mode* on page 2-15 to undo specific swaps. Rollback swaps can be scheduled for execution according to normal Optimizer policy.

Optimizer Logs and Swap History

To get a list of all swaps that have been completed, use the following command:

```
symoptmz show -swap_hist
```

To retrieve the Optimizer activity log, including swaps that have been completed, use the following command:

```
symoptmz read -log_type RUNTIME [-start <date_time>]
[stop <date_time>]
```

To retrieve the Optimizer error log, use the command:

```
symoptmz read -log_type ERROR [-start <date_time>]
[stop <date_time>]
```

The log that is returned includes DOS line-feed characters that can be removed using sed or another text editor:

```
cat opt_log.txt | sed s/^M$// > cleaned_opt_log.txt
```

In a UNIX shell, the ^M should be typed using **CTRL-V CTRL-M**.

In addition, the `symoptmz read` command can be used to determine why a swap was not performed.

Command File General Parameters

General Optimizer parameters are set using the following syntax in the command file:

```
set control_parms [start_mode=<AUTO | MANUAL>,]
[min_perf_period=<min_perf>,]
[workload_period=<workload>,]
[swap_mode=<AUTO | USER_OK>,]
[max_simult_swaps=<max_simult>,]
[swap_rate=<max_swaps>];
```

The following table describes each parameter.

Table 2-2 Parameters in the Command File

Parameter	Description
<i>start_mode</i>	Determines whether Optimizer is enabled or disabled whenever Optimizer is launched, such as after the service processor is rebooted.
<i>swap_mode</i>	Controls whether Optimizer should automatically swap volumes (AUTO) as soon as it finds swaps that would improve performance. If Optimizer is enabled in AUTO swap mode, each day it will make up to the number of swaps specified by the <i>swap_rate</i> parameter, under control of the swap time window settings. If Optimizer is enabled in USER_OK swap mode, it will generate lists of swap suggestions approximately once an hour, and then wait for the user to approve the swap list before proceeding.
<i>min_perf_period</i>	Specifies the amount of samples required initially before a recommendation will be made. You should make sure that the values you specify are long enough (usually a week) for Optimizer to establish a good characterization of your typical workloads. This parameter is expressed in hours. Keep in mind that the Optimizer statistics database holds about 14 days worth of data.
<i>workload_period</i>	Specifies how far back in time Optimizer should consider when the optimization algorithm is run. Be careful not to make this value too large or you may include data that is so old it is no longer representative of your current workloads. This parameter is expressed in hours.
<i>max_simult_swaps</i>	Controls how many swaps Optimizer can perform simultaneously (up to four). The actual value should reflect the number of volumes that will be swapped simultaneously. The acceptable values are from two (a single pair swapped) to eight (four pairs swapped) Optimizer swaps.
<i>swap_rate</i>	Sets the maximum number of swaps that Optimizer is allowed to make in a single day. This parameter is only relevant if <i>swap_mode</i> is set to AUTO. Note that reflects the total number of volumes to be swapped in a day. A value of 24 would allow 12 pairs of volumes to be swapped in a day.

Example

The following command file sets up Optimizer to analyze data from the previous seven days and to start figuring out swap suggestions after three days of collecting data. It also sets Optimizer to User Approved mode and sets the maximum number of simultaneous swaps to eight (four pairs of hypervolumes).

```
set control_parms start_mode=AUTO,  
min_perf_period=72,  
workload_period=168  
swap_mode=USER_OK,  
max_simult_swaps=8,  
swap_rate=50;
```

Quality of Service

This chapter describes the control operations that manage the Quality of Service (QoS) on devices in your storage environment.

- ◆ Overview3-2
- ◆ Setting QoS Copy Parameters and Values.....3-3
- ◆ Setting QoS LRU Parameters and Values3-5
- ◆ Viewing QoS Parameters3-6

Overview

Quality of Service (QoS) allows you more flexibility in managing your Symmetrix system's performance. By increasing the response time for specific copy operations on selected devices, you can increase the overall performance of the other Symmetrix devices.

The QoS (Quality of Service) SYMCLI feature, available with Enginuity version 5x66 and higher, allows you to adjust the data transfer (copy) pace on specified devices or device groups for certain operations.

Starting with Enginuity version 5x67, the contention for cache access can be quality of service managed by the least recently used (LRU) ring partitions in the Symmetrix cache.

Setting QoS Copy Parameters and Values

Using SYMCLI, you can set QoS parameters and values for specified devices managing the copy pace (priority) and the LRU assignments.

Copy Pace to Certain Devices

Different copy pace (priority) levels can be set on certain types of devices. Possible copy pace levels on the devices are 0, the highest, to 10, the lowest. Zero is the default.

To specify which copy operations you are adjusting the pace for, there are five copy operation types to choose from:

- ◆ BCV—TimeFinder™ copy operations
- ◆ RDF—SRDF® copy operations
- ◆ BCS—Backup Control System (Fastrax) copy operations
- ◆ MIR—Mirror copy operations
- ◆ CLONE—Clone (previously called Snap) copy operations

To set a copy pace to 1 during BCV operations for the standard devices in group `ProdDB`, enter:

```
symqos set BCV pace 1 -g ProdDB -nobcv
```

or:

```
symqos set BCV priority 1 -g ProdDB -nobcv
```

Note that `pace` and `priority` in the syntax are interchangeable nouns, which can be used to identify and set the same parameter.

For BCV operations, to set the BCV copy pace value to 1 on logical device `BCV002` of device group `ProdDB`, enter:

```
symqos set BCV pace 1 -g ProdDB BCV002
```

For RDF operations, in device group `ProdDB`, to set the copy pace level to 2 for all STD and BCV devices, enter:

```
symqos set RDF pace 2 -g ProdDB
```

For RDF operations, to set the RDF copy pace value to 1 on just BCV devices of device group `ProdDB`, enter:

```
symqos set RDF pace 1 -g ProdDB -bcv
```

For recovery operations, to set the BCS copy pace value to 2 on all STD and BCV devices of device group `ProdDB`, enter:

```
symqos set BCS pace 2 -g ProdDB
```

To set a copy pace to 1 during copy operations for mirror device `DEV021` in group `ProdDB`, enter:

```
symqos set MIR pace 1 -g ProdDB DEV021
```

Setting QoS LRU Parameters and Values

Using SYMCLI, you can set QoS parameters and values for specified devices managing LRU cache I/O operations.

LRU Cache I/O Management

Starting with Enginuity version 5x67, the contention for cache access can be managed by the least recently used (LRU) ring partitions in the Symmetrix cache. You can manage the cache strategy utilizing LRU rings (groups) and assigning certain devices or device groups to them.

To assign all devices of device group `ProdDB` to LRU group 2, enter:

```
symqos -g ProdDB set LRU 2
```

Or, to assign all devices of device group `ProdDB` to an LRU group named `GROUP_2`, enter:

```
symqos -g ProdDB set LRUname GROUP_2
```

Using the `-lru` option of `symcfg list`, you can list the cache-slot allocation and allocation percentage of a specified LRU cache management group number. You can list all the LRUs if you specify `ALL`.

Using the `-reset` option of `symqos`, you can reset the LRU assignment back to the default LRU. For example:

```
symqos reset LRU LdevName
```

Viewing QoS Parameters

Using SYMCLI, you can get a list of QoS parameters and values for specified devices.

For example, to get a list of copy pace levels for devices 010 through 0B2 in Symmetrix 010000002334, enter:

```
symqos list -sid 2334 -range 10:B2
```

Note that if you omit the `-sid` option, the specified range of devices will apply to all Symmetrix units.

Note that in the preceding example, concerning the range option, the *SymdevStart* value will default to zero if you omit the value as follows:

```
symqos list -sid 2334 -range :B2
```

This will increase the range to include all devices between 000 and 0B2. If you omit the *SymdevEnd* value, the range will list all devices found beyond the *SymdevStart* value on the specified Symmetrix unit as follows:

```
symqos list -sid 2334 -range 10:
```

To list the LRU assignments of devices 00C through 013 to the various LRU cache management groups, enter:

```
symqos list -lru -sid 346 -range 00C:013
```

To get the copy pace levels concerning just the BCV devices of device group ProdDB, enter:

```
symqos -g ProdDB query -bcv
```

To get the copy pace levels concerning both the BCV and STD devices of device group ProdDB, enter:

```
symqos -g ProdDB query
```

To get the copy pace levels concerning just the STD devices of device group ProdDB, enter:

```
symqos -g ProdDB query -nobcv
```

To view the LRU assignment of all devices in a device group, enter:

```
symqos -g ProdDB query -lru
```

To assign all devices of device group `ProdDB` to LRU group 2, enter:

```
symqos -g ProdDB set LRU 2
```


C

- code examples, Quality of Service 3-6
- command file
 - mandatory actions to perform 2-15
- conventions 1-viii
- Copy operation
 - for Clone 3-3
- copy operation
 - for BCS 3-3
 - for BCV 3-3
 - for mirrors 3-3
 - for RDF 3-3

O

- Optimizer
 - self-tuning features 1-1, 1-2, 2-1

Q

- Quality of Service
 - code examples 3-6
 - Enginuity version requirement 3-5
 - parameters for 3-6
 - setting the copy pace level 3-3

S

- SYMCLI
 - control commands 1-2
- symoptmz 1-3
- symqos 1-3

