

SANtricity[®] Storage Manager Command Line Interface and Script Commands

Programming Guide for Version 9.10

ES1436-0-E1, First Edition



Proprietary Rights Notice

This document contains proprietary information of Engenio Information Technologies, Inc. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of Engenio. Any product(s) described herein is/are a licensed product of Engenio.

Document Description

Document ES1436-0-E1, First Edition. July 2004

This document describes the command line interface and script command software for SANtricity[®] Storage Manager version 9.10, and will remain the official reference source for all revisions and releases of this product until rescinded by an update.

Disclaimer

It is the policy of Engenio Information Technologies to improve products as new technology, components, software, and firmware become available. Engenio reserves the right to make changes to any products herein at any time without notice. All features, functions, and operations described herein may not be marketed by Engenio in all parts of the world. In some instances, photographs and figures are of equipment prototypes. Therefore, before using this document, consult your Engenio representative for information that is applicable and current. **ENGENIO DOES NOT ASSUME ANY RESPONSIBILITY OR LIABILITY FOR THE USE OF ANY PRODUCT(S) DESCRIBED HEREIN EXCEPT AS EXPRESSLY AGREED TO IN WRITING BY ENGENIO.**

License Restriction

The purchase or use of an Engenio product does not convey a license under any patent, copyright, trademark, or other intellectual property right of Engenio or third parties.

Copyright Notice

© 2004. Engenio Information Technologies, Inc. All rights reserved.

Trademark Acknowledgments

Engenio, the Engenio design, and HotScale are trademarks, and SANtricity Storage Manager and SANshare are registered trademarks of Engenio Information Technologies, Inc. All other brand and product names may be trademarks of their respective companies.

Conventions

The following conventions have been used throughout this book.

Definitions of Safety Notices

DANGER indicates an imminently hazardous situation that will result in death or severe personal injury.

WARNING indicates a potentially hazardous situation that could result in death or severe personal injury.

CAUTION indicates a potentially hazardous situation that could result in moderate or minor personal injury.

Definitions of Informational Notices

CAUTION indicates a potentially hazardous situation that could result in data loss (or other interruption) or equipment damage.

IMPORTANT indicates information or criteria that is necessary to perform a procedure correctly.

NOTE indicates a concept that will be clarified or a maintenance tip that will be presented.

Revision Record

Edition or Revision	Date	Affected Pages or Remarks
First Edition	July 2004	New Book.

Part Number: ES1436-0-E1

Contents

ABOUT THE COMMAND LINE INTERFACE

How to Use the Command Line Interface.....	1-3
Formatting Considerations	1-9
Detailed Error Reporting	1-11
Exit Status	1-12
Usage Examples	1-13

ABOUT THE SCRIPT COMMANDS

Script Command Structure	2-3
Script Command Synopsis	2-5
Recurring Syntax Elements.....	2-7
Usage Guidelines.....	2-10
Adding Comments to a Script File.....	2-11

CONFIGURING A STORAGE ARRAY

Configuration Concepts.....	3-2
Configuring a Storage Array	3-10
Modifying Your Configuration	3-21

USING THE SNAPSHOT PREMIUM FEATURE

How Snapshot Works.....	4-2
Creating a Snapshot Volume.....	4-4
Changing Snapshot Volume Settings.....	4-11
Stopping, Restarting, and Deleting a Snapshot Volume	4-12

USING THE REMOTE VOLUME MIRRORING PREMIUM FEATURE

How Remote Volume Mirroring Works	5-2
Creating a Remote Volume Mirror Pair	5-8
Changing Remote Volume Mirror Settings	5-14
Suspending and Resuming a Mirror Relationship.....	5-15
Removing a Mirror Relationship.....	5-17
Deleting a Primary or Secondary Volume	5-18
Disabling the Remote Volume Mirror Feature	5-19

Deactivating the Remote Volume Mirror Feature.....	5-20
Interaction with Other Features	5-21

USING THE VOLUME COPY PREMIUM FEATURE

How Volume Copy Works	6-2
Creating a Volume Copy	6-7
Viewing Volume Copy Properties	6-10
Changing Volume Copy Settings	6-11
Re-Copying a Volume.....	6-13
Stopping a Volume Copy	6-14
Removing Copy Pairs.....	6-15
Interaction with other Features.....	6-16

MAINTAINING A STORAGE ARRAY

Routine Maintenance.....	7-2
Performance Tuning	7-6
Troubleshooting and Diagnostics	7-9
Recovery Operations	7-16

SCRIPT COMMANDS

Command Formatting Rules	8-2
Firmware Compatibility levels.....	8-4
Commands Listed Alphabetically	8-5
Commands Listed by Function.....	8-127

EXAMPLES OF INFORMATION RETURNED BY SHOW COMMANDS

Show Storage Array	A-2
Show Controller NVSRAM	A-17
Show Drive.....	A-20
Show Drive Channel Status.....	A-26
Show Volumes	A-27

EXAMPLE SCRIPT FILES

Configuration Script Example 1	B-2
Configuration Script Example 2.....	B-4

List of Tables

Table 1-1. Command Name Conventions	1-4
Table 1-2. Command Line Parameters	1-5
Table 2-1. Configuration and Management Operations	2-1
Table 2-2. Object Types and Identifiers	2-3
Table 2-3. Script Command General Form	2-5
Table 2-4. Recurring Syntax Elements	2-7
Table 3-1. Maximum Number of Physical Disks	3-3
Table 3-2. Maximum Volume Values	3-6
Table 3-3. RAID Level Configurations	3-7
Table 3-4. Maximum Number of Host Ports	3-8
Table 3-5. Cache Flush Modifier Values	3-25
Table 4-1. Snapshot Volume Components	4-2
Table 4-2. Snapshot Volume Commands	4-3
Table 4-3. Snapshot Volume Parameters	4-7
Table 5-1. Remote Volume Mirror Commands	5-6
Table 6-1. Volume Copy Commands	6-5
Table 7-1. Storage Array Performance Information	7-6
Table 7-2. Storage Array Support Data	7-9
Table 7-3. Storage Array RLS Baseline Data	7-13
Table C-1. Deprecated Commands	C-1
Table C-2. Deprecated Keywords	C-7

About This Book

This book is a programming guide that describes the command line interface and the script commands for SANtricity Storage Manager 9.10. It explains how to use the command line interface and the script commands to configure, operate, and maintain a storage array. This book documents all the script commands, explains the purpose of the commands, shows the complete syntax, and defines the parameters for the commands. This book assumes that the user has a knowledge of basic storage area network (SAN) hardware and installation skills. Read the SANtricity Storage Manager Product Release Notes for any updated information regarding hardware, software, or firmware products that may not be covered in this document.

Intended Readers

This book is intended for system operators, system administrators, and service personnel who are responsible for operating, maintaining, troubleshooting, and servicing a storage array. Users must be familiar with computer system operation, maintenance, and repair. In addition, they should understand disk array, RAID, network and Fibre Channel technologies. The reader must have a basic knowledge of SAN hardware functionality (controllers, drives, hosts) and SAN cabling techniques; and must understand disk array, Redundant Array of Independent Disks (RAID), network, and Fibre Channel technologies

Related Publications

Command Line Interface and Script Commands Quick Reference

Web Address

For web sites related to the products in this publication, please see the Product Release Notes.

Chapter 1

About the Command Line Interface

The command line interface (CLI) is a software tool that enables storage array installers, developers, and engineers to configure and monitor storage arrays. Using the command line interface, you can issue commands from an operating system prompt, such as the DOS C:\ prompt, a Linux path, or a Solaris path. Each command performs a specific action for managing a storage array or returning information about the status of a storage array. You can enter individual commands or you can run script files when you need to perform operations more than once (such as installing the same configuration on several storage arrays). The command line interface enables you to load a script file from a disk and run the script file. The command line interface provides a way to run storage management commands on more than one network storage array. You can employ the command line interface in both installation sites and development environments.

The command line interface gives you direct access to a script engine that is a utility in the SANtricity Storage Manager. The script engine runs commands that enable you to configure and manage storage arrays. The script engine reads the commands, or runs a script file, from the command line and performs the operations instructed by the commands.

NOTE You can also access the script engine using the Enterprise Management Window; however, if you access the script engine using the Enterprise Management Window you can edit or run script commands on only one storage array in the script window. You can open a script window for each storage array in your configuration and run commands in each window. Using the command line interface, you can run commands on more than one storage array from a single command line.

You can use the command line interface to:

- Directly access the script engine and run script commands.
- Create script command batch files to be run on multiple storage arrays when you need to install the same configuration on different storage arrays.
- Run script commands on an in-band managed storage array, an out-of-band managed storage array, or a combination of both.
- Display configuration information about the network storage arrays.
- Add storage arrays to and remove storage arrays from the management domain.

-
- Perform automatic discovery of all storage arrays attached to the local subnet.
 - Add or delete Simple Network Management Protocol (SNMP) trap destinations and email alert notifications.
 - Specify the mail server and sender email address or SNMP server for alert notifications.
 - Display the alert notification settings for storage arrays currently configured in the Enterprise Management Window.
 - Direct the output to a standard command line display or to a named file.

How to Use the Command Line Interface

The commands you run on the command line interface (CLI commands) provide access to the script engine, specify the storage array to receive the script commands, and set operation environment parameters.

A CLI command consists of the following elements:

- The term **SMcli**
- Storage array identifier
- Parameters
- Script commands

The general form of a CLI command is:

```
SMcli host parameters script-commands;
```

SMcli invokes the command line interface, **storage array** is the name or IP address of the storage array, **parameters** are CLI parameters that define the environment and purpose for the command, and **script-command** is one or more script commands or the name of a script file containing script commands. (The script commands are the storage array configuration commands. Chapter 2, “About the Script Commands,” presents an overview of the script commands. Chapter 8, “Script Commands,” provides definitions, syntax, and parameters for the script commands.)

Usage Notes

If you enter **SMcli** and a storage array name, but do not specify CLI parameters, script commands, or a script file, the command line interface runs in interactive mode. Interactive mode enables you to run individual commands without prefixing the commands with **SMcli**. In interactive mode you can enter a single command, see the results, and enter the next command without typing the complete **SMcli** string. Interactive mode is useful for determining configuration errors and quickly testing configuration changes.

If you enter **SMcli** without any parameters or an incorrect parameter, the script engine returns usage information.

CLI Commands

This section lists the CLI commands you can use to identify storage arrays, specify passwords, add storage arrays to configuration files, specify communications parameters, enter individual script configuration commands, or specify a file containing script configuration commands. Table 1-1 on page 1-4 lists the conventions used in the general form of the CLI command.

.....

Table 1-1 Command Name Conventions

Convention	Definition
a b	alternative (“a” or “b”)
<i>italicized-words</i>	terminals
[...] (square brackets)	zero or one occurrence
{...} (curly braces)	zero or more occurrences
(a b c)	choose only one of the alternatives
bold	terminals

The general forms of the CLI commands, showing the parameters and terminals used in each command, are listed below. Table 1-2 lists definitions for the parameters shown in the general form of the CLI commands.

```
SMcli host-name-or-IP-address [host-name-or-IP-address] [-c "command; {command2;}"] [-n storage-array-name | -w WWN] [-o outputfile] [-p password] [-e] [-S]
```

```
SMcli host-name-or-IP-address [host-name-or-IP-address] [-f scriptfile] [-n storage-array-name | -w WWN] [-o outputfile] [-p password] [-e] [-S]
```

```
SMcli -n storage-array-name | -w WWN [-c "command; {command2;}"] [-o outputfile] [-p password] [-e] [-S]
```

```
SMcli -n storage-array-name | -w WWN [-f scriptfile] [-o outputfile] [-p password] [-e] [-S]
```

```
SMcli -n storage-array-name | -w WWN [-o outputfile] [-p password] [-e] [-S]
```

```
SMcli (-a | -x) email:email-address [-n storage-array-name | -w WWN | -h host-name | -r (inband_sa | outofband_sa)] [-S]
```

```
SMcli (-a | -x) email:email-address host-name-or-IP-address [host-name-or-IP-address] [-n storage-array-name | -w WWN | -h host-name | -r (inband_sa | outofband_sa)] [-S]
```

```
SMcli (-a | -x) trap:community,host-name-or-IP-address [-n storage-array-name | -w WWN | -h host-name | -r (inband_sa | outofband_sa)] [-S]
```

```
SMcli (-a | -x) trap:community,host-name-or-IP-address host-name-or-IP-address [host-name-or-IP-address] [-n storage-array-name | -w WWN | -h host-name | -r (inband_sa | outofband_sa)] [-S]
```

```

.....

SMcli -d [-w] [-i] [-s] [-v] [-S]

SMcli -m host-name-IP-address -F email-address [-S]

SMcli -A [host-name-or-IP-address [host-name-or-IP-address]]
[-S]

SMcli -X (-n storage-array-name | -w WWN | -h host-name)

SMcli -?

```

Command Line Parameters

Table 1-2 Command Line Parameters

Parameter	Definition
<i>hostname or IP address</i>	<p>You can specify either the host name or the IP address (xxx.xx.xx.xx) of an in-band managed storage array or an out-of-band managed storage array.</p> <ol style="list-style-type: none"> 1. If you are managing a storage array using a host through in-band storage management, you must use the <code>-n</code> or <code>-w</code> parameter if more than one storage array is connected to the host. 2. If you are managing a storage array using out-of-band storage management through the ethernet connection on each controller, you must specify the <i>host-name-or-IP-address</i> of the controllers. 3. If you have previously configured a storage array in the EMW, you can specify the storage array by its user-supplied name using the <code>-n</code> parameter. 4. If you have previously configured a storage array in the EMW, you can specify the storage array by its world wide name using the <code>-w</code> parameter.
<code>-A</code>	<p>Use this parameter to add a storage array to the configuration files. If you do not follow the <code>-A</code> parameter with a <i>host-name-or-IP-address</i>, auto-discovery scans the local subnet for storage arrays.</p>

Parameter	Definition
-a	<p>Use this parameter to add an SNMP trap destination or an email address alert destination.</p> <ul style="list-style-type: none"> • When adding an SNMP trap destination, the SNMP community is automatically defined as the community name for the trap and the <i>host</i> is the IP address or DNS host name of the system to which the trap should be sent. • When adding an email address for an alert destination the <i>email-address</i> is the email address where you want the alert message to be sent.
-c	<p>Use this parameter to indicate you are entering one or more script commands to run on the specified storage array. Each command must be terminated by a semicolon (;).</p> <p>You cannot place more than one -c parameter on the same command line. You can include more than one script command after the -c parameter.</p>
-d	<p>Use this parameter to display the contents of the script configuration file. The format of the file contents is:</p> <pre>storage-Array-name host-name1 host-name-2</pre>
-e	<p>Use this parameter to run the commands without performing a syntax check first.</p>
-f	<p>Use this parameter to specify a file name containing script commands you want to run on the specified storage array. (This parameter is similar to the -c parameter in that both are intended for running script commands. The -c parameter runs individual script commands. The -f parameter runs a file of script commands.)</p> <p>By default, any errors encountered when running the script commands in a file are ignored, and the file continues to run. To override this behavior, use the <code>set session errorAction=stop</code> command in the script file .</p>
-F	<p>Use this parameter to specify the email address from which all alerts will be sent.</p>
-h	<p>Use this parameter to specify the host name that is running the SNMP agent to which the storage array is connected. Use this parameter with the -a and -x parameters.</p>

Parameter	Definition
-i	<p>Use this parameter to display the IP address of the known storage arrays. Use this parameter with the <code>-d</code> parameter. The format of the file content is:</p> <pre>storage-Array-name IP-address-1 IP-address-2</pre>
-m	<p>Use this parameter to specify the host name or IP address of the email server from which email alert notifications will be sent.</p>
-n	<p>Use this parameter to specify the name of the storage array on which you want to run the script commands. This name is optional when you use a “host-name” or “IP address;” however, if you are using the in-band method for managing the storage array, you must use the <code>-n</code> parameter if more than one storage array is connected to the host at the specified address.</p> <p>The storage array name is required when the <i>host-name</i> or <i>IP address</i> is not used; however, the name of the storage array configured for use in the EMW (that is, listed in the configuration file) must not be a duplicate name of any other configured storage array.</p>
-o	<p>Use this parameter with the <code>-c</code> or <code>-f</code> parameters to specify a file name for all output text that is a result of running the script commands. If you do not specify an output file, the output text will go to stdout. All output from commands that are not script commands is sent to stdout, regardless of whether this parameter is or is not set.</p>
-p	<p>Use this parameter to specify the password for the storage array on which you want to run commands. A password is not necessary under the following conditions:</p> <ol style="list-style-type: none"> 1. A password has not been set on the storage array. 2. The password is specified in a script file you are running. 3. You specify the password using the <code>-c</code> parameter and the <code>set session password=string-literal</code> command.
-r	<p>Use this parameter with the <code>-a</code> or <code>-x</code> parameters to specify the name of an organizer node. The name of an organizer node can be either <i>outofband_sa</i> (out-of-band storage array organizer node) or <i>inband_sa</i> (in-band storage arrays organizer node). The <code>-r</code> parameter enables you to set or change the alert notifications for all storage arrays under each organizer node.</p>

Parameter	Definition
-S	<p>Use this parameter to suppress informational messages describing command progress that appear when running script commands. (Suppressing informational messages is also called “silent mode.”) This parameter suppresses the following messages:</p> <ul style="list-style-type: none"> • Performance syntax check. • Syntax check complete. • Executing script. • Script execution complete. • SMcli completed successfully.
-s	Use this parameter with the <code>-d</code> parameter to display the alert settings in the configuration file.
-v	Use this parameter with the <code>-d</code> parameter to display the current global status of the known devices in a configuration file.
-w	<p>Use this parameter to specify the world wide name (WWN) of the storage array. This parameter is an alternate to the <code>-n</code> parameter. Use the <code>-w</code> parameter with the <code>-d</code> parameter of display the WWNs of the known storage arrays. The format of the file content is:</p> <pre>storage-Array-name world-wide-name IP- address-1 IP-address-2</pre>
-X	Use this parameter to delete a storage array from a configuration.
-x	Use this parameter to remove an SNMP trap destination or and email address alert destination. The <i>community</i> is the SNMP community name for the trap and the <i>host</i> is the IP address or DNS host name of the system to which you want the trap sent.
-?	Use this parameter to display usage information about the CLI commands.

Formatting Considerations

A double quote symbol (“”) used as part of a name or label requires special consideration when you run the CLI and script commands on a Windows operating system. The following bullet items explain how to use double quotes in names while running CLI and script commands on a Windows operating system.

- When double quotes are part of a name or argument, you must insert a backslash (\) before each double quote character. For example:

```
-c "set storageArray userLabel=\"Engineering\";"
```

where Engineering is the storage array name. A second example is:

```
-n "\"My\" Array"
```

where “My” Array is the name of the storage array.

- You cannot use the double quote character (“”) as part of a character string (also called *string literal*) within a script command. For example, you cannot enter the following string to set the storage array name to “Finance” Array:

```
-c "set storageArray userLabel=\"\"Finance\"Array\";"
```

In a UNIX operating system the delimiters around names or labels are single quotes. The UNIX versions of the previous examples are:

```
-c `set storageArray userLabel="Engineering";`
```

```
-n `"My" Array`
```

Because the backslash character (\) is used with double quotes in a Windows operating system, the backslash requires special consideration when used as part of a name or label.

- Use the backslash character (\) as typed, unless you want to use it immediately before a double quote character (“”). Use three backslashes before the double quote character to have the backslash printed when you use the `-n`, `-o`, `-f`, or `-p` parameters. For example, to specify a storage array named Array\, enter:

```
-n "Array\\\""
```

- To use a backslash character (\) as part of a string literal within a script command, the same rules as in the previous bullet item apply except that you need to use five backslashes. For example, to change the name of a storage array to Array, enter the following string:

```
-c "set storageArray userLabel="Array\\\\\\\";"
```

.....

A Windows operating system requires that you insert a caret (^) before each special script character. Special characters are ^, &, |, <, >

- Insert a caret before each special script character when used with the `-n`, `-o`, `-f`, and `-p` parameters. For example, to specify storage array CLI&CLIENT enter the following string:

```
-n "CLI^&CLIENT"
```

- Insert three carets (^^^) before each special script character when used within a script command string literal. For example, to change the name of a storage array to FINANCE&PAYROLL, enter the following string:

```
-c "set storageArray userLabel=\"FINANCE^^^&PAYROLL\";"
```

Detailed Error Reporting

Error data collected from an error encountered by the script engine is written to a file. Detailed error reporting under the script engine works as follows:

- If the script engine must abnormally end running CLI and script commands, error data will be collected and saved before the script engine terminates.
- The script engine saves the error data by writing the data to a standard file name (similar to a UNIX “core” file).
- The script engine automatically saves the data to a file; special command line options are not required to save the error data.
- User interaction is not required to save the error data to a file.
- The script engine has no provision to avoid over-writing an existing version of the file containing error data.

Exit Status

After you run a CLI command or CLI and script command, status indicating the success of the operation defined by the command is displayed. The status are:

0 - terminated without error.

1 - terminated with error. Information about the error is also displayed.

Usage Examples

The following examples show how to enter CLI commands on a command line. The examples show the syntax, form, and, in some examples, script commands. Examples are shown for both Windows and UNIX systems. Note that the usage for the `-c` parameter varies depending on your operating system. On Windows operating systems, the script command following the `-c` must be enclosed in double quotation marks (“”). On UNIX systems, the script command following the `-c` must be enclosed in single quotation marks (‘’). (Refer to Chapter 8, “Script Commands,” for descriptions of the script commands used in the following examples.)

1. This example shows how to change the name of a storage array. The original name of the storage array is **Payroll Array**. The new name will be **Finance Array**.

Windows operating system:

```
SMcli ICTSANT -n "Payroll Array" -c "set storagearray  
userlabel=\"Finance Array\";"
```

UNIX operating system:

```
SMcli ICTSANT -n 'Payroll Array' -c 'set storagearray  
userlabel="Finance Array";'
```

2. This example shows how to delete an existing volume and create a new volume on a storage array. The existing volume name is **Stocks & Bonds**. The new volume name will be **Finance**. The controller host names are **finance1** and **finance2**. The storage array is protected, requiring the password **TestArray**.

Windows operating system:

```
SMcli finance1 finance2 -c "set session  
password=\"TestArray\"; delete volume[\"Stock^^&  
Bonds\"]; create volume driveCount[3] RAIDLEVEL=3  
capacity=10GB userLabel=\"Finance\"; show storageArray  
healthStatus;"
```

UNIX operating system:

```
SMcli finance1 finance2 -c 'set session  
password="TestArray"; delete volume["Stock & Bonds"];  
create volume driveCount[3] RAIDLEVEL=3 capacity=10GB  
userLabel="Finance"; show storageArray healthStatus;'
```

-
3. This example shows how to run commands in a script file named **scriptfile.scr** on a storage array named **Example**. The **-e** parameter causes the file to run without checking syntax. Running a script file without checking syntax enables the file to run more quickly; however, the file might not run correctly because syntax for a command might be incorrect.

```
SMcli -n Example -f scriptfile.scr -e
```

4. This example shows how to run commands in a script file named **scriptfile.scr** on a storage array named **Example**. In this example the storage array is protected by the password **My Array**. Output, as a result of commands in the script file, will go to file *output.txt*.

Windows operating system:

```
SMcli -n Example -f scriptfile.scr -p "My Array" -o  
output.txt
```

UNIX operating system:

```
SMcli -n Example -f scriptfile.scr -p 'My Array' -o  
output.txt
```

5. This example shows how to display all storage arrays in the current configuration. The command in this example returns the host name of each storage array.

```
SMcli -d
```

If you want to know the IP address of each storage array in the configuration, add the **-i** parameter to the command.

```
SMcli -d -i
```

Chapter 2

About the Script Commands

You can use the script commands to configure and manage a storage array. The script commands are distinct from the CLI commands; however, you enter the script commands using the command line interface. You can enter individual script commands or you can run a file of script commands. When you enter an individual script command, you include it as part of a CLI command. When you run a file of script commands, you include the file name as part of a CLI command. The script commands are processed by a script engine that

- Verifies command syntax
- Interprets the commands
- Converts the commands to the appropriate protocol-compliant commands
- Passes the commands to the storage array

At the storage array, the script commands are run by the storage array controllers.

The script engine and script commands support the storage array configuration and management operations listed in Table 2-1.

Table 2-1 Configuration and Management Operations

Operation	Activities
Cache Configuration	Controlling all cache parameters, both at storage array and individual volume level
Volume, Volume Group Configuration	Creating, deleting, setting reconstruction priority control, labeling, setting drive composition when creating volumes, setting media scan control
Drive Configuration	Assigning hot spares
Controller Configuration	Defining volume ownership, changing mode settings
General storage array Configuration	Resetting a configuration to defaults, labeling, checking health status, setting time of day, clearing event log, setting media scan rate
NVSRAM Configuration	Downloading, modifying the user configuration region at the bit and byte level, displaying NVSRAM values

Operation	Activities
Product Identification	Retrieving module profile display data
Battery Management	Setting battery installation date
Firmware Management	Downloading fimware

Script Command Structure

All script commands have the following structure:

```
command operand-data { statement-data }
```

where *command* identifies the action to be performed, *operand-data* represents the objects associated with a storage array you want to configure or manage, and *statement-data* provides the information needed to perform the command.

The syntax for *operand-data* is:

```
(object-type | allobject-types | [qualifier] (object-type  
[identifier] {object-type identifier}) | object-types  
[identifier-list]))
```

An object can be identified four ways: object type, *all* parameter, square brackets, or a list of identifiers. Use an object type when the command is not referencing a specific object. The *all* parameter means all objects of the specified type in the storage array (for example, allVolumes). To perform a command on a specific object, use square brackets to identify the object (for example, volume[engineering]). Specify a subset of objects with a list of identifiers in square brackets (for example, volumes[sales engineering marketing]). A qualifier is necessary if you want to include additional information to describe the objects. Table 2-2 lists the object type and identifiers associated with the object types.

Table 2-2 Object Types and Identifiers

Object Type	Identifier
controller	a or b
drive	tray id and slot id
driveChannel	drive channel identifier
host	user label
hostChannel	host channel identifier
hostGroup	user label
hostPort	user label
remote mirror	primary volume user label
snapshot	volume user label
storageArray	N/A
tray	tray id
volume	volume user label or volume WWN (set command only)

.....

Object Type	Identifier
volumeCopy	target volume and optionally the source volume user labels
volumeGroup	volume group number

Statement data is in the form of **attribute=value** (raidLevel=5), an **attribute name** (batteryInstallDate), or an **operation name** (redundancyCheck).

Script Command Synopsis

Because you can use the script commands to define and manage the different aspects of a storage array (such as host topology, drive configuration, controller configuration, volume and volume group definitions), the actual number of commands is extensive. The commands, however, fall into general categories that are reused when you apply the commands to the different aspects of a storage array. Table 2-3 lists the general form of the script commands and provides a definition of each command.

Table 2-3 Script Command General Form

Syntax	Description
<code>activate object {statement-data}</code>	Sets up the environment so that an operation can take place or performs the operation if the environment is already properly set up.
<code>autoConfigure storageArray{statement-data}</code>	Automatically creates a configuration based on parameters specified in the command.
<code>check object {statement-data}</code>	Starts an operation to report on errors in the object. This is a synchronous operation.
<code>clear object {statement-data}</code>	Discards the contents of some attribute of an object. This is a destructive operation that cannot be reversed.
<code>create object {statement-data}</code>	Creates an object of the specified type.
<code>deactivate object {statement-data}</code>	Removes the environment for an operation.
<code>delete object</code>	Deletes a previously created object.
<code>diagnose object {statement-data}</code>	Runs a test and displays the results.
<code>disable object {statement-data}</code>	Prevents a feature from operating.
<code>download object {statement-data}</code>	Transfers data to the storage array or hardware associated with the storage array.
<code>enable object {statement-data}</code>	Allows a feature to operate.
<code>recopy object {statement-data}</code>	Restarts a volume copy operation using an existing volume copy pair. You can change attributes before the operation is restarted.

Syntax	Description
<code>recover object {statement-data}</code>	Recreates an object from saved configuration data and the statement attributes. (Similar to the <code>create</code> command.)
<code>recreate object {statement-data}</code>	Restarts a snapshot operation using an existing snapshot volume. You can change attributes before the operation is restarted.
<code>remove object {statement-data}</code>	Removes a relationship from between objects.
<code>repair object {statement-data}</code>	Repairs errors found by the <code>check</code> command.
<code>reset object {statement-data}</code>	Returns the hardware or object to an initial state.
<code>resume object</code>	Starts a suspended operation. The operation begins where it left off when suspended.
<code>revive object</code>	Forces the object from the failed to the optimal state. Use this command only as part of an error recovery procedure.
<code>save object {statement-data}</code>	Writes information about the object to a file.
<code>set object {statement-data}</code>	Changes object attributes. All changes are completed when the command returns.
<code>show object {statement-data}</code>	Displays information about the object.
<code>start object {statement-data}</code>	Starts an asynchronous operation. You can stop some operations after they have started. You can query the progress of some operations.
<code>stop object {statement-data}</code>	Stops an asynchronous operation.
<code>suspend object {statement-data}</code>	Stops an operation. You can then restart the suspended operation and it continues from the point where it was suspended.

Recurring Syntax Elements

Recurring syntax elements are a general category of variables and parameters you can use in one or more script commands. The recurring syntax is used in the general definitions of the script commands that are listed in “Chapter 8, Script Commands.” Table 2-4 lists the recurring syntax and the syntax values that you can use with the syntax. The conventions listed in Table 1-1 on page 1-4 define the meaning of the syntax values.

Table 2-4 Recurring Syntax Elements

Recurring Syntax	Syntax Value
<i>raid-level</i>	(0 1 3 5)
<i>repository-raid-level</i>	(1 3 5)
<i>capacity-spec</i>	<i>integer-literal</i> [KB MB GB TB Bytes]
<i>segment-size-spec</i>	<i>integer-literal</i>
<i>boolean</i>	(TRUE FALSE)
<i>user-label</i>	<i>string-literal</i>
<i>user-label-list</i>	<i>user-label</i> { <i>user-label</i> }
<i>create-raid-vol-attr-value-list</i>	<i>create-raid-volume-attribute-value-pair</i> { <i>create-raid-volume-attribute-value-pair</i> }
<i>create-raid-volume-attribute-value-pair</i>	capacity = <i>capacity-spec</i> owner =(<i>a</i> <i>b</i>) readAheadMultiplier = <i>integer-literal</i> segmentSize = <i>integer-literal</i>
<i>trayId</i>	(0 - 99)
<i>slot-id</i>	(1 - 32)
<i>port-id</i>	(0 - 127)
<i>drive-spec</i>	<i>trayId</i> , <i>slotId</i>
<i>drive-spec-list</i>	<i>drive-spec</i> { <i>drive-spec</i> }
<i>tray-id-list</i>	<i>trayId</i> { <i>trayId</i> }
<i>hex-literal</i>	0x <i>hexadecimal-literal</i>
<i>volume-group-number</i>	<i>integer-literal</i>
<i>filename</i>	<i>string-literal</i>
<i>error-action</i>	(stop continue)

Recurring Syntax	Syntax Value
<i>drive-channel-identifier</i>	(1 2 3 4)
<i>drive-channel-identifier-list</i>	<i>drive-channel-identifier</i> { <i>drive-channel-identifier</i> }
<i>host-channel-identifier</i>	(a1 a2 b1 b2)
<i>drive-type</i>	(fibre SATA PATA)
<i>feature-identifier</i>	(storagePartition2 storagePartition4 storagePartition8 storagePartition16 storagePartition64 storagePartitionMax snamshot remoteMirror volumeCopy)
<i>repository-spec</i>	<i>instance-based-repository-spec</i> <i>count-based-repository-spec</i>
<i>instance-based-repository-spec</i>	repositoryRAIDLevel = <i>repository-raid-level</i> repositoryDrives =(<i>drive-spec-list</i>) [trayLossProtect = <i>boolean</i> ¹] repositoryVolumeGroup = <i>volume-group-number</i> repositoryFreeCapacityArea = <i>integer-literal</i> ² Specify either repositoryRAIDLevel with repositoryDrives , or repositoryVolumeGroup . Do not specify RAID level or drives with volume group.
<i>count-based-repository-spec</i>	repositoryRAIDLevel = <i>repository-raid-level</i> repositoryDriveCount = <i>integer-literal</i> [driveType = <i>drive-type</i> ³] [trayLossProtect = <i>boolean</i> ⁴]
WWN	<i>string-literal</i>
<i>nvsram-offset</i>	<i>hex-literal</i>
<i>host-type</i>	<i>string-literal</i> <i>integer literal</i>
<i>nvsram-byte-setting</i>	<i>nvsram-value</i> -> <i>0xhexadecimal</i> <i>integer-literal</i>
<i>nvsram-bit-setting</i>	<i>nvsram-mask</i> , <i>nvsram-value</i> -> <i>0xhexadecimal</i> , <i>0xhexadecimal</i> <i>integer-literal</i>
<i>ip-address</i>	[0 - 255]. [0 - 255]. [0 - 255]. [0 - 255]
<i>autoconfigure-vols-attr-value-list</i>	<i>autoconfigure-vols-attr-value-pair</i> { <i>autoconfigure-vols-attr-value-pair</i> }

Recurring Syntax	Syntax Value
<i>autoconfigure-vols-attr-value-pair</i>	driveType = <i>drive-type</i> ⁵ raidLevel = <i>raid-level</i> volumeGroupWidth = <i>integer-literal</i> volumeGroupCount = <i>integer-literal</i> volumesPerGroupCount = <i>integer-literal</i> ⁶ hotSpareCount = <i>integer-literal</i> segmentSize = <i>segment-size-spec</i> readAheadMultiplier = <i>integer-literal</i>
<i>create-volume-copy-attr-value-list</i>	<i>create-volume-copy-attr-value-pair</i> { <i>create-volume-copy-attr-value-pair</i> }
<i>create-volume-copy-attr-value-pair</i>	copyPriority =(highest high medium low lowest) targetReadOnlyEnabled = <i>boolean</i>
<i>recover-raid-volume-attr-value-list</i>	<i>recover-raid-volume-attr-value-pair</i> { <i>recover-raid-volume-attr-value-pair</i> }
<i>recover-raid-volume-attr-value-pair</i>	owner =(a b) readAheadMultiplier = <i>integer-literal</i>
<i>cache-flush-modifier-setting</i>	immediate , 0 , .25 , .5 , .75 , 1 , 1.5 , 2 , 5 , 10 , 20 , 60 , 120 , 300 , 1200 , 3600 , infinite

¹For tray loss protection to work, each physical disk in a volume group must be on a separate module. If you set `trayLossProtect=TRUE` and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect=FALSE`, the storage array will perform operations, but the volume group you create may not have tray loss protection.

²To determine if a free capacity area exists, issue the `show volumeGroup` command.

³The default physical disk (drive type) is “fibre” (Fibre Channel).

⁴If you set `trayLossProtect` to true, the storage array will return an error if the controller firmware cannot find drives that will enable the new volume group to have tray loss protection. If you set `trayLossProtect` to false, the storage array will perform the operation even if it means the volume group may not have tray loss protection.

⁵The `driveType` parameter is not required if only one type of physical disk is in the storage array. If you use the `driveType` parameter, you must also use the `hotSpareCount` and `volumeGroupWidth` parameters. If you do not use the `driveType` parameter, the configuration defaults to Fibre Channel physical disks.

⁶The number of equal capacity volumes per volume group.

Usage Guidelines

- You must end all commands with a semicolon (;).
- You can enter more than one command on a line, but you must separate each command with a semicolon (;).
- You must separate each base command and its associated primary and secondary parameters with a space.
- The script engine is not case sensitive. You can enter commands using upper case, lower case, or mixed case.
- Add comments to your scripts to make it easier for you and future users to understand the purpose of the script commands. (For information on how to add comments, refer to “Adding Comments to a Script File.”)

.....

Adding Comments to a Script File

You can add comments to a script file in three ways.

1. The script engine interprets any text typed after two forward slashes (//) until an end of line character is reached. If the script engine does not find an end of line character in the script after processing a comment, an error message is displayed and the script operation is terminated. This error commonly occurs when a comment is placed at the end of a script and you have forgotten to press the *Enter* key.

```
// Deletes the existing configuration.  
set storageArray resetConfiguration=true;
```

2. The script engine interprets any text type between /* and */ as a comment. If the script engine does not find both a beginning and ending comment notation, an error message is displayed and the script operation is terminated.

```
/* Deletes the existing configuration */  
set storageArray resetConfiguration=true;
```

3. Use the **show** statement to embed comments in a script file that you want to display while the script file is running. The text you want to display must be enclosed by quotation marks ("").

```
show "Deletes the existing configuration";  
set storageArray resetConfiguration=true;
```

.....

Chapter 3

Configuring a Storage Array

This chapter explains how to run script commands from the command line to create a volume from a group of physical disks and configure a RAID storage array. This chapter assumes that you have a basic understanding of RAID concepts and terminology. Before you begin to configure your storage array, become familiar with the concepts of physical disks, volume groups, volumes, host groups, hosts, and controllers.

Configuring a RAID storage array requires caution and planning to ensure that you define the correct RAID level and configuration for your storage array. The main purpose in configuring a storage array is to create volumes, addressable by the hosts, from a collection of physical disks. The commands described in this chapter enable you to set up and run a RAID storage array. Additional commands are also available to provide you with more control and flexibility; however, many of these commands require a deeper understanding of the firmware as well as various structures that need to be mapped. Use all of the CLI and script commands with caution.

The following sections in this chapter show some, but not all of the CLI and script commands. The purpose of showing these commands is to explain how you can use the commands to configure a storage array; however, the presentation does not explain all possible usage and syntax for the commands. (For complete definitions of the commands, including syntax, parameters, and usage notes, refer to Chapter 8, “Script Commands.”)

This chapter contains examples of CLI and script command usage. The command syntax used in the examples is for a host running a Microsoft operating system. As part of the examples, the complete C:\ prompt and DOS path for the commands are shown. Depending on your operating system the prompt and path construct will vary; however, the CLI and script command syntax is the same for all operating systems.

Configuration Concepts

When you configure a storage array, you are organizing physical disks into a logical structure that provides storage capacity and data protection so that one or more hosts can safely store data in the storage array. This section provides definitions of the physical and logical components required to organize the physical disks into a configuration and describes how the components relate to each other.

Controllers

All storage arrays have one or two controllers. The controllers are circuit board assemblies that provide the interface between the hosts and the storage array, and manage the data flow between the hosts and the physical disks. In general, each controller has a processor for performing control operations, NVSRAM for storing the firmware code that operates the storage array, and the buses along which the data flows.

The controllers are located in a command module. The command module provides two positions for controllers: slots A and B. The script commands identify each controller by the slot in which the controller is installed. If a storage array has only one controller, the controller must be in slot A.

The controllers are connected to the hosts and the physical disks through interface modules. Each interface module is a circuit board assembly installed in the command module. The host side has four interface modules. Each module has two connectors to support up to eight hosts. Each interface module is identified by the controller slot to which the interface module is connected. Two interface modules are connected to slot A and two interface modules are connected slot B. When viewed from the rear of the command module, the host interface modules are numbered from left to right **a1, b1, a2, b2**. The script commands identify the host channels using these identifiers.

The physical disk side has four interface modules. Each module has two connectors to support redundant drive loop configurations. Each module represents a single channel to the physical disks. When viewed from the rear of the command module, the physical disk interface modules are numbered from left to right **4, 3, 2, 1**. The script commands use these numbers to identify the physical disk channels.

Physical Disks

The physical disks provide the actual storage of the data sent to the storage array. The physical disks are mounted in a drive module. (The drive module is an enclosure that, in addition to the physical disks, contains power supplies, fans, environmental cards, and other supporting components.)

Physical disks are located in a storage array by tray ID and slot ID. Tray ID values are 0 to 99. Tray ID values are set during installation by switches on the back of the modules.

.....

The slot ID is the physical disk position in the drive module. Slot ID values range from 1 to 32. A drive module can contain either 10 or 14 physical disks. The total number of physical disks in a storage array depends on the model of the controller and the type of drive module. Table 3-1 lists, by controller number and drive module capacity, the maximum number of physical disks in a storage array.

Table 3-1 Maximum Number of Physical Disks

Controller	10 Drive Module	14 Drive Module
2880		14
2882		112
4884	220	224
5884	220	224

Hot Spares

A hot spare is a physical disk that acts as a standby in the event a physical disk containing data fails. The hot spare is a physical disk that has not been assigned to a particular volume group and as such can be used in any volume group. You can use the hot spare feature with RAID 1, 3 or 5. If a physical disk in a volume group fails, the controllers automatically replace the failed physical disk with a hot spare. The controllers use redundancy data to reconstruct the data from the failed physical disk onto the hot spare. To be most effective, the physical disk you assign as a hot spare must have a capacity equal to or greater than the capacity of the largest physical disk in the storage array. The hot spare must be the same type of physical disk as the physical disk that failed (that is, a SATA hot spare cannot replace a Fibre Channel physical disk).

You can assign physical disks to act as hot spares manually or have the script commands automatically assign hot spares. If you manually assign a physical disk to be a hot spare, you must identify the physical disk by tray ID and slot ID. When you let the script commands automatically assign hot spares, you need to enter the number of hot spares you want in the storage array. The maximum number of Fibre Channel hot spare physical disks per storage array is 15.

Volume Groups

A volume group is a set of physical disks logically grouped together by the controllers in a storage array. After you create a volume group you can then create one or more volumes in the volume group. A volume group is identified by a sequence number defined by the firmware when you created the volume group.

To create a volume group you must define the capacity and RAID level.

.....

Capacity is the size of the volume group and is determined by the number of physical disks you assign to the volume group. You can use only unassigned physical disks to create a volume group. (In this programming guide, the storage space on unassigned physical disks constitutes the *unconfigured capacity* of a storage array.)

Free capacity is a contiguous region of unassigned capacity in a designated volume group. Before you create a new volume in a volume group, you need to know the free capacity space so that you can determine the size of the volume.

The number of physical disks you can include in a volume group is constrained by the capacity of each physical disk. The following list relates the capacity of individual physical disks to the number of physical disks you can include in a volume group.

- 18 and 36 GB disks, 30 disks maximum in the volume group
- 73 GB disks, 28 disks maximum in the volume group
- 180 GB disks, 11 disks maximum in the volume group

A 73 or 180 GB physical disk will not report exactly 73 or 180 GB as its capacity. You cannot have a combination of large physical disks in a volume group that will exceed the two terabyte size limitation.

You can determine the volume group size by multiplying the maximum number of physical disks in the volume group by the capacity of the smallest physical disk in the volume group.

The RAID level is the level of data protection you want to define for your storage array. The RAID level you choose affects storage capacity. When you are configuring your storage array you need to take into consideration this compromise between data protection and storage capacity. In general, the more protection you need the less storage capacity will be available in your storage array. (For an explanation of the RAID levels you can define using the script commands, refer to “RAID Levels” on page 3-6.)

Volumes

A volume is a logical component (object) that is the basic structure created on the storage array to store data. A volume is a contiguous subsection of a volume group that is configured to meet application needs for data availability and I/O performance. The storage management software administers a volume as if the volume is one “drive” for data storage. Volumes are identified by names or labels users choose. The volume name can be any combination of alpha numeric characters with a maximum length of 30 characters.

The script commands support the following types of volumes:

- *Standard volume* – a standard volume is a logical structure that is the principal type of volume for data storage. A standard volume is the most common type of volume in a storage array.

-
- *Access volume* – a factory configured volume in a storage area network (SAN) environment used for communication between the storage management software and the storage array controller. The access volume uses a LUN address and consumes 20 MB of storage space. The 20 MB of access volume storage space is not available for data storage.

NOTE You need to use the access volume only for out-of-band managed storage arrays.

- *Snapshot volume* – a logical point-in-time image of another volume. A snapshot volume is the logical equivalent of a complete physical copy; however, it is not an actual, physical copy. Instead, the firmware tracks only the data blocks that are overwritten and copy those blocks to a repository volume.
- *Repository volume* – a special volume in the storage array created as a resource for a snapshot volume. A repository volume contains snapshot data and copy-on-write data for a particular snapshot volume.
- *Base volume* – a standard volume from which you create a snapshot volume. The term “base volume” is used only to show the relationship between a standard volume from which you are taking the point-in-time image and a snapshot volume.
- *Primary volume* – a standard volume in a remote volume mirror relationship. The primary volume accepts host data transfers and stores application data. When you first create the mirror relationship, data from the primary volume is copied in its entirety to the associated secondary volume.
- *Secondary volume* – a standard volume in a remote volume mirror relationship that maintains a mirror (or copy) of the data from its associated primary volume. The secondary volume remains unavailable to host applications while mirroring is underway. In the event of a disaster or catastrophic failure of the primary site, a system administrator can promote the secondary volume to a primary role.
- *Mirror repository volume* – a special volume in a remote volume mirror configuration created as a resource for each controller in both the local and remote storage array. The controller stores mirroring information on this volume, including information about remote writes that are not yet complete. A controller can use this information to recover from controller resets and accidental power shut-down of the storage arrays.

NOTE Snapshot volume and remote volume mirroring are premium features that you must activate before you can use the features. For more information about snapshots volumes refer to Chapter 4, “Using the Snapshot Premium Feature.” For more information about Remote Volume Mirroring, refer to Chapter 5, “Using the Remote Volume Mirroring Premium Feature.”

The number and capacity of the volumes in your storage array depends on the type of controller in the storage array. Table 3-2 lists the maximum values for the volumes in a storage array by specific controllers.

Table 3-2 Maximum Volume Values

	2880	2882	4884	5884
Maximum Volumes Per Storage Array	1024	1024	2048	2048
Maximum Volumes Per Volume Group	256	256	256	256
Maximum Volume Size	2TB minus 12GB	2TB minus 12GB	2TB minus 12GB	2TB minus 12GB
Maximum Physical Disks Per Volume Group	30	30	30	30
Total Remote Mirrors	0	32	64	64

RAID Levels

The RAID level defines a storage architecture in which the storage capacity on the physical disks in a volume group is separated into two parts: part of the capacity is used to store the user data and the remainder is used to store redundant information about the user data. The RAID level you choose determines how user and redundancy data are written and retrieved from the physical disks. Using the script commands, you can define four RAID levels: 0, 1, 3, and 5. Each level provides different performance and protection features. RAID 0 provides the fastest storage access but does not provide any redundant information about the stored data. RAID levels 1, 3, and 5 write redundancy information to the physical disks to provide fault tolerance. The redundancy information might be a copy of the data or an error-correcting code derived from the data. In a RAID level 1, 3, or 5 configuration, if a physical disk fails the redundancy information can be used to reconstruct the lost data. Regardless of the RAID level you choose, you can configure only one RAID level across each volume group. All redundancy information for a volume group is stored within the volume group. Table 3-3 lists the RAID levels and describes the configuration capabilities of each level.

Table 3-3 RAID Level Configurations

RAID Level	Configuration
0	Non-redundant striping mode. Use this level for high performance needs. RAID 0 does not provide any data redundancy. RAID 0 stripes data across all physical disks in the volume group. If a single physical disk fails, all associated volumes fail and all data is lost. RAID 0 is suited for non critical data. It is not recommended for high availability needs.
1	Striping mirroring mode. RAID 1 uses physical disk mirroring to make an exact copy from one physical disk to another physical disk. A minimum of two physical disks are required, one for the user data and one for the mirrored data. RAID 1 offers high performance and the best data availability. Data is written to two physical disks simultaneously. If one physical disk in a pair fails, the system can instantly switch to the other physical disk without any loss of data or service. Only half of the physical disks in the volume group are available for user data. If a single physical disk fails in a RAID 1 volume group, all associated volumes become degraded, but the mirror physical disk provides access to the data. RAID 1 can survive multiple physical disk failures as long as no more than one failure occurs per mirrored pair. If a physical disk pair fails, all associated volumes fail, and all data is lost.
3	High bandwidth mode. RAID 3 stripes both user data and redundancy data (in the form of parity) across the physical disks. The equivalent of the capacity of one physical disk is used for the redundancy data. RAID 3 works well for large data transfers in applications such as multimedia or medical imaging that write and read large sequential chunks of data. If a single physical disk fails in a RAID 3 volume group, all associated volumes become degraded, but the redundancy data enables the data to be reconstructed. If two or more physical disks fail, all associated volumes fail, and all data is lost.
5	High I/O mode. RAID 5 stripes both user data and redundancy data (in the form of parity) across the physical disks. The equivalent of the capacity of one physical disk is used for the redundancy data. RAID 5 works well for multi-user environments such as data bases or file system storage where typical I/O size is small and there is a high proportion or read activity. If a single physical disk fails in a RAID 5 volume group, all associated volumes become degraded, but the redundancy data enables the data to be reconstructed. If two or more physical disks fail, all associated volumes fail, and all data is lost.

Hosts

A host is a computer attached to the storage array for the purpose of accessing the volumes in the storage array. The host is attached to the storage array through host ports, which are host bus adapter circuit boards. You can define specific volume-to-LUN mappings to an individual host or assign the host to a host group that shares access to one or more

.....

volumes. Hosts are identified by names or labels users choose. The host name can be any combination of alphanumeric characters with a maximum length of 30 characters.

In addition to a host name, some script commands require you to identify a host by its “type.” A host type identifies the operating system under which the host is running (such as Windows, Solaris, or Linux). Specifying the host type enables the controllers in the storage array to adapt their behavior (such as LUN reporting and error conditions) to the operating system of the host sending the information. Host types are identified by a label or index number generated by the firmware.

Host Groups

A host group is a topological element that you can define if you want to designate a collection of hosts that will share access to the same volumes. A host group is a logical entity. Host groups are identified by names or labels users choose. The host group name can be any combination of alpha numeric characters with a maximum length of 30 characters.

Host Port

A host port is the physical connection between the host and the storage array. The host port is a physical connector on a host bus adapter. The host bus adapter is a circuit board installed in the host. The host bus adapter can have one or more host ports. If the host bus adapter has more than one host port, each host port will have a unique ID. Each host port is identified by a unique, 16-byte world wide name (WWN).

When you first power on a storage array, the storage management software automatically detects the host ports. Initially, all detected host ports belong to a default group. You can use script commands to identify the WWNs on a storage array and, if you choose, change them. If you move a host port, you must remap any volume-to-LUN mappings. Access to your data is lost until you remap the volumes.

The maximum number of host ports you can logically define for your storage array depends on the type of controller in the storage array. Table 3-4 lists the maximum number of host ports you can define.

Table 3-4 Maximum Number of Host Ports

2880	2882	4884	5884
256	256	512	512

Logical Unit Number

In the context of the CLI and script commands, a logical unit number (LUN) is a unique value used to identify the volumes in a storage array. The hosts identify the volumes they want to access using the LUN values. When you create a volume, LUN values are assigned by the firmware, or you can assign LUN values when you enable the SANshare storage partitioning premium feature. A volume can have only one LUN and can be mapped to only one host or host group. Each host has unique addressing capability; that is, when more than one host accesses a storage array each host might use the same LUN to access different volumes. The LUNs might be the same, but the volumes are different. If you are mapping to a host group, the LUN you specify must be available on every host in the host group.

Configuring a Storage Array

When you configure a storage array, you want to maximize data availability by ensuring the data is quickly accessible while maintaining the highest level of data protection possible. The speed by which a host can access data is affected by the RAID level for the volume group and settings for segment size, cache size, and cache read ahead multiplier. Data protection is determined by the RAID level, hardware redundancy (such as global host spares), and software redundancy (such as the remote volume mirroring and snapshot volume premium features).

In general, you configure a storage array by defining a volume group and associated RAID level, defining the volumes, and defining which hosts have access to the volumes. This section explains how to use the script commands to perform the general steps to create a configuration from an array of physical disks.

Determining What Is on Your Storage Array

Even when you create a configuration on a storage array that has never been configured, you still need to determine the hardware and software features that are to be included with the storage array. When you are configuring a storage array that has an existing configuration, you must ensure that your new configuration does not inadvertently alter the existing configuration, unless you are reconfiguring the entire storage array. For example, consider the case where you want to create a new volume group on unassigned physical disks; before you create a new volume group you must determine which physical disks are available. The commands described in this section enable you to determine the components and features in your storage array.

The command that returns general information about the storage array is the `show storageArray` command. This command returns information about the components and properties of your storage array including:

- Detailed profile of the components and features in the storage array
- Battery age
- Default host type (which is the current host type)
- Other host types that you can select
- Hot spare locations
- Identifiers for enabled features
- Logical and physical component profiles
- Time to which both controllers are set
- Controller that currently owns each volume in the storage array

.....

To return the most information about the storage array, run the `show storageArray` command with the `profile` parameter. An example of the complete CLI and script command running on a Microsoft operating system is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storagearray profile;"
```

This example identifies the storage array by the dummy IP address 123.45.67.89. You can also identify the storage array by name.

The `show storageArray profile` command returns detailed information about the storage array. The information covers several screens on a display. You may need to increase the size of your display buffer to see all the information. Because this information is so detailed, you may want to save the output to a file. To save the output to a file, enter the command as shown in the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storagearray profile;" -o c:\folder\storarrayprofile.txt
```

IMPORTANT When you are writing information to a file, the script engine does not check to determine if the filename already exists. If you choose the name of a file that already exists, the script engine will write over the information in the file without warning.

“Show Storage Array” on page A-2 shows the type of information returned. When you save the information to a file, you can use the information as a record of your configuration and as an aid during recovery.

To return a brief list of the storage array features and components, use the `summary` parameter. The command is similar to:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storagearray summary;"
```

The following example shows the type of information returned by the `show storagearray` command with the `summary` parameter.

.....

PROFILE FOR STORAGE ARRAY: example9 (x/xx/xx 9:58:32 AM)

SUMMARY-----
Number of controllers: 1
Number of volume groups: 2
Total number of volumes (includes an access volume): 3 of 2048 used
 Number of standard volumes: 2
 Number of access volumes: 1
 Number of snapshot repositories: 0
Number of snapshot volumes: 0
Number of copies: 0
Number of drives: 9
Supported drive types: Fibre (9)
Total hot spare drives: 0
 Standby: 0
 In use: 0
Access volume: LUN 7 (see Mappings section for details)
Default host type: Linux (Host type index 6)
Current configuration
 Firmware version: PkgInfo 96.10.21.00
 NVSRAM version: N4884-610800-001
Pending configuration
 Staged firmware download supported?: No
 Firmware version: Not applicable
 NVSRAM version: Not applicable
 Transferred on: Not applicable
NVSRAM configured for batteries?: Yes
Start cache flushing at (in percentage): 80
Stop cache flushing at (in percentage): 80
Cache block size (in KB): 4
Media scan frequency (in days): Disabled
Failover alert delay (in minutes): 5
Feature enable identifier: 1234567891011121314151617181ABCD

The summary information is also returned as the first section of information when you use the `profile` parameter.

The following `show` commands return information about the specific components of a storage array. The information returned by each of these commands is the same as the information returned by the `show storagearray profile` command, but constrained to the specific component.

- `show controller`
- `show drives`
- `show drivechannels stats`
- `show storageArray hostTopology`
- `show storageArray lunmappings`
- `show allvolumes`
- `show volumeGroup`
- `show volume reservations`

.....

Other commands that return information about a storage array are:

- `show storageArray autoConfigure`
- `show controller NVSRAM`
- `show remoteMirror candidates`
- `show storageArray unreadableSectors`
- `show volumeCopy sourceCandidates`
- `show volumeCopy targetCandidates`
- `show volume performanceStat`

For descriptions of the show commands, including examples of the information returned by each command, refer to Chapter 8, “Script Commands.” Other commands can help you learn about your storage array. To see a list of the commands, refer to “Commands Listed by Function” on page 8-127. These commands are organized by the storage array activities the commands support (for example, volume commands, host commands, module commands, and others).

Clearing the Configuration

If you want to create a completely new configuration on a storage array that already has an existing configuration, use the `clear storageArray configuration` command to remove existing configuration information. This command deletes all configuration information, including all volume group, volume, and hot spare definitions, from the controller memory.

CAUTION Potential storage array configuration damage. As soon as you run this command, the existing storage array configuration is deleted.

The command syntax is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "clear storageArray configuration;"
```

Do not run the `clear storageArray configuration` command to create additional volumes in a pre-existing configuration; this will destroy the pre-existing configuration. Use the `clear storageArray configuration` command only when creating a new configuration.

Using the Auto Configure Command

The `autoConfigure storageArray` command creates the volume groups on a storage array, the volumes in the volume groups, and the hot spares for the storage array. When you use the `autoConfigure storageArray` command, you define the following parameters:

- Type of physical disks (Fibre, SATA, or PATA)
- RAID level
- Number of physical disks in a volume group
- Number of volume groups
- Number of volumes in each volume group
- Number of hot spares
- Size of each segment on the physical disks
- Read ahead multiplier

After you define these parameters, the storage management software creates the volume groups, volumes, and hot spares. The controllers assign volume group and volume numbers as they are created. After the storage management software creates the initial configuration, you can use the `set volume` command to define volume labels.

Before you run the `autoConfigure storageArray` command, run the `show storageArray autoConfigure` command. The `show storageArray autoConfigure` returns a list of parameter values the storage management software will use to automatically create a storage array. If you would like to change any of the parameter values, you can do so by entering new values for the parameters when you run the `autoConfigure storageArray` command. If you are satisfied with the parameter values the `show storageArray autoConfigure` command returns, run the `autoConfigure storageArray` command without new parameter values.

The general syntax for the `autoConfigure storageArray` command is:

```
autoConfigure storageArray [driveType=(fibre | SATA | PATA)
raidLevel=(0 | 1 | 3 | 5) | volumeGroupWidth=numberOfDrives
| volumeGroupCount=numberOfVolumeGroups |
volumesPerGroupCount=numberOfVolumesPerGroup |
hotSpareCount=numberOfHotspares |
segmentSize=segmentSizeValue |
readAheadMultiplier=multiplierValue]
```

NOTE All parameters are optional. You can use one or all parameters as needed to help define your configuration.

When you use the `autoConfigure storageArray` command without specifying the number of volume groups, the firmware determines how many volumes and volume groups to create. The firmware will create from one volume group and volume to the maximum number the storage array can support. When you specify the number of volume groups, the firmware creates only that number of volume groups. When you create more than one volume group, all the volume groups have the same number of physical disks and the same number of volumes.

The `volumeGroupWidth` parameter defines the number of unassigned physical disks you want to use for each new volume group.

The `volumeGroupCount` parameter defines the number of new volume groups you want in the storage array.

The `volumesPerGroupCount` parameter defines the number of volumes you want in each volume group.

The `hotSpareCount` parameter defines the number of hot spares you want in each volume group.

The `segmentSize` parameter defines the amount of data, in kilobytes, the controller writes on a single physical disk in a volume before writing data on the next physical disk. The smallest units of storage are data blocks. A data block stores 512 bytes of data. The size of a segment determines how many blocks it contains. An 8 KB segment holds 16 data blocks. A 64 KB segment holds 128 data blocks. For optimal performance in a multi-user data base or file system storage environment, set the segment size to minimize the number of physical disks needed to satisfy an I/O request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests.

If the volume is for a single user with large I/O requests (such as multimedia), performance is maximized when a single I/O request can be serviced with a single data stripe. A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data storage. In this environment, multiple physical disks are used for the same request, but each physical disk is accessed only once.

The `readAheadMultiplier` parameter defines the multiplier value used to determine how many additional data blocks are read into cache. Cache read ahead enables the controller to copy additional data blocks into cache while it is reading and copying host requested data blocks from a physical disk into cache. This increases the chance that a future request for data can be fulfilled from the cache, which improves the speed with which data is accessed. Choosing a higher cache read ahead multiplier may increase the cache hit percentage; however, transfer time might also increase.

Cache read ahead multiplier values range from 0 to 65535. Use small values for random size data files. Use large values for large files with sequential I/O requests such as

.....

multimedia applications. If you do not want to enable cache read ahead, set the `readAheadMultiplier` parameter to 0.

After you have finished creating the volume groups and volumes using the `autoConfigure storageArray` command, you can further define the properties of the volumes in a configuration using the `set volume` command. (For an explanation of how to further define your configuration, refer to “Modifying Your Configuration” on page 3-21.)

Example of the Auto Configuration Command

```
c:\...\sm9\client>smcli 123.45.67.89 -c "autoConfigure
storageArray driveType=fibre raidLevel=5 volumeGroupWidth=8
volumeGroupCount=3 volumesPerGroupCount=4 hotSpareCount=2
segmentSize=8 readAheadMultiplier=1;"
```

The command in this example will create a storage array configuration using Fibre Channel physical disks set to RAID level 5. Three volume groups will be created, each volume group consisting of eight physical disks, which are configured into four volumes. The storage array will have two hot spares. The segment size for each volume will be 8 KB. The cache read ahead multiplier will be “1,” which causes one additional data block to be written into cache.

Using the Create Volume Command

The `create volume` command enables you to create new storage array volumes in three ways.

- Create a new volume while simultaneously creating a new volume group to which you assign the physical disks.
- Create a new volume while simultaneously creating a new volume group to which the storage management software assigns the physical disks.
- Create a new volume in an existing volume group.

You do not need to assign the entire capacity of the volume group to the volume.

Creating Volumes with User Assigned Physical Disks

When you create a new volume and assign the physical disks you want to use, the storage management software creates a new volume group. The controller firmware assigns a volume group number to the new volume group. The general syntax for this command is:

.....

```
create volume drives=(trayID1,slotID1...trayIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5) userLabel=volumeName
[capacity=volumeCapacity owner=(a | b)
readAheadMultiplier=multiplierValue
segmentSize=segmentSizeValue] [trayLossProtect=(TRUE |
FALSE) ]
```

NOTE capacity, owner, readAheadMultiplier, segmentSize, and trayLossProtect are optional parameters. You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

The `userLabel` parameter is the name that you want to give to the volume. The volume name can be any combination of alphanumeric characters with a maximum length of 30 characters. You must enclose the volume name with double quotes (“”).

The `drives` parameter is a list of the physical disks that you want to use for the volume group. Enter the tray ID and slot ID of each physical disk that you want to use. Enclose the list in parentheses, separate the tray ID and slot ID of a physical disk by a comma, separate each tray ID and slot ID pair by a space. For example:

```
(1,1 1,2 1,3 1,4 1,5)
```

The `capacity` parameter defines the size of the volume. You do not have to assign the entire capacity of the physical disks to the volume. You can later assign any unused space to another volume.

The `owner` parameter defines the controller to which you want to assign the volume. If you do not specify a controller, the controller firmware will determine the volume owner.

The `readAheadMultiplier` and `segmentSize` parameters are the same as those described for the `autoConfigure storageArray` command. Descriptions of these parameters start on page 3-15.

The `trayLossProtect` parameter defines tray loss protection for the volume group. (For a description of how tray loss protection works, refer to “Tray Loss Protection” on page 3-19.)

Example of Creating Volumes with User Assigned Physical Disks

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create volume
drives=(1,1 1,2 1,3 2,1 2,2 2,3) raidLevel=5
userLabel=\"Engineering 1\" capacity=20GB owner=a
readAheadMultiplier=5 segmentSize=128;"
```

The command in this example will automatically create a new volume group and a volume with the name “Engineering 1.” The volume group will have a RAID level of five (RAID 5). The command will use six physical disks to construct the volume group. The physical disks will have a total volume capacity of 20 GB. If each physical disk has a capacity of

.....

18 GB the total capacity of all the assigned disks is 108 GB. Because only 20 GB is assigned to the volume, 88 GB remains available for other volumes a user can later add to this volume group. The cache read ahead multiplier will be five, which causes five additional data block to be written into cache. The segment size for each volume will be 128 KB. Tray loss protection is set to true, preventing any operations to physical disks in the drive module is the drive module fails. Hot spares have not been created for this new volume group. A user will need to create hot spares after running this command.

Creating Volumes with Software Assigned Physical Disks

If you choose to let the storage management software assign the physical disks when you create the volume, you need only to specify the number of physical disks you want to use. The storage management software then assigns the physical disks. The controller firmware assigns a volume group number to the new volume group. The general syntax for manually creating volume groups and volumes is:

```
create volume driveCount=numberOfDrives raidLevel=(0 | 1 | 3
| 5) userLabel=volumeName [driveType=(fibre | SATA | PATA)]
[capacity=volumeCapacity | owner=(a | b) |
readAheadMultiplier=multiplierValue |
segmentSize=segmentSizeValue] [trayLossProtect=(TRUE |
FALSE) ] )
```

NOTE driveType, capacity, owner, readAheadMultiplier, segmentSize, and trayLossProtect are optional parameters. You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

This command is similar to the previous `create volume` command in which users assign the physical disks. The difference between this command and the previous one is that this version of the command requires only the number and the type of physical disks you want to use in the volume group. You do not need to enter a list of physical disks. All other parameters are the same. Tray loss protection is performed differently when the storage management software assigns the physical disks than when a user assigns the physical disks. (For an explanation of the difference, refer to “Tray Loss Protection.”)

Example of Creating Volumes with Software Assigned Physical Disks

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create volume
driveCount=6 raidLevel=5 userLabel=\"Engineering 1V\"
capacity=20GB owner=a readAheadMultiplier=5 segmentSize=128;
```

The command in this example will create the same volume as the example for the previous `create volume` command in which a user assigns the physical disks. The difference is that a user will not know which physical disks are assigned to this volume group.

Creating Volumes in an Existing Volume Group

If you want to add a new volume to an existing volume group, use the following command:

```
create volume volumeGroup=volumeGroupName  
userLabel=volumeName [freeCapacityArea=freeCapacityValue |  
capacity=volumeCapacity | owner=(a | b) |  
readAheadMultiplier=multiplierValue |  
segmentSize=segmentSizeValue]
```

NOTE *freeCapacityArea*, *capacity*, *owner*, *readAheadMultiplier*, and *segmentSize* are optional parameters. You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

The *volumeGroup* parameter is the number of the volume group in which you want to create a new volume. If you do not know the volume group numbers on the storage array, you can use the `show allVolumes summary` command to display a list of the volumes and the volume groups to which the volumes belong.

The *userLabel* parameter is the name that you want to give to the volume. The volume name can be any combination of alphanumeric characters with a maximum length of 30 characters. You must enclose the volume name with double quotes (“ ”).

The *freeCapacityArea* parameter defines the free capacity area to use for the volume. If a volume group has several free capacity areas, you can use this parameter to identify which area to use for volume creation. You do not have to assign the entire capacity of the physical disks to the volume. You can later assign any unused space to another volume.

The *userLabel*, *capacity*, *owner*, *readAheadMultiplier*, and *segmentSize* parameters are the same as in the previous versions of the `create volume` command.

Tray Loss Protection

The *trayLossProtect* parameter is a boolean switch that you set to turn on or turn off tray loss protection. For tray loss protection to work, each physical disk in a volume group must be on a separate module. The way in which tray loss protection works depends on the method you choose to assign the physical disks for a volume group.

When you assign the physical disks, if you set *trayLossProtect*=TRUE and have selected more than one physical disk from any one module, the storage array will return an error. If you set *trayLossProtect*=FALSE, the storage array will perform operations, but the volume group you create will not have tray loss protection.

When the controller firmware assigns the physical disks, if *trayLossProtect*=TRUE the storage array will error the function if the controller firmware cannot provide physical

.....

disks that will result in the new volume group having tray loss protection. If `trayLossProtect=FALSE` the storage array will perform the operation even if it means the volume group may not have tray loss protections.

Tray loss protection is not valid when creating volumes on existing volume groups.

Modifying Your Configuration

For most configurations, after you have created your initial configuration using the `autoConfigure storageArray` or `create volume` commands you will need to modify the properties of your configuration to ensure it performs to meet the requirements for data storage. Use the “set” commands to modify a storage array configuration. This section describes how to modify the following properties:

- Controller clocks
- Storage array password
- Storage array host type
- Storage array cache
- Global hot spares

Setting the Controller Clocks

If you need to synchronize the clocks on the controllers with the host, you can do so using the `set storageArray time` command. Running this command helps ensure that event timestamps written by controllers to the event log match event timestamps written to the host log files. The controllers remain available during synchronization. An example of the command is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set storagearray  
time;"
```

Setting the Storage Array Password

The `set storageArray` command enables you to define a password for a storage array. The general form of the command is:

```
set storageArray password= "password"
```

The `password` parameter defines a password for the storage array. Passwords provide added security to a storage array to help reduce the possibility of implementing destructive commands.

CAUTION Implementing destructive commands can cause serious damage, including data loss.

Unless you define a password for the storage array, all script commands can be run. A password protects the storage array from any command the controllers consider destructive. Destructive commands are any command that can change the state of the storage array, such as volume creation, cache modification, reset, delete, rename, or

.....

change commands. If you have more than one storage array in a storage configuration, each storage array has a separate password. Passwords can have a maximum length of 30 alphanumeric characters. You must enclose the password in double quotes (“”). An example of using the `set storageArray` command to define a password is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set storagearray
password="1a2b3c4d5e";"
```

Setting the Storage Array Host Type

The `set storageArray` command enables you to define the default host type. The general form of the command is:

```
set storageArray defaultHostType=(hostTypeName |
hostTypeIdentifier)
```

The `defaultHostType` parameter defines how the controllers in the storage array will communicate with the operating system on undefined hosts that are connected to the storage array SAN. This parameter defines the host type only for storage array data I/O activities; this parameter does not define the host type for the management station. The operating system can be either Microsoft, Linux, or Solaris. For example, if you set the `defaultHostType` to Linux, the controller will communicate with any undefined host if the undefined host is running a Linux operating system. Typically, you would need to change the host type only when you are setting up the storage array. The only time you may need to use this parameter is if you need to change how the storage array behaves relative to the hosts that are connected to it.

Before you can define the default host type, you need to determine what host types are connected to the storage array. To return information about host types connected to the storage array, you can use the `show storageArray` command with the `defaultHostType` parameter or `hostTypeTable` parameter. This command returns a list of the host types with which the controllers can communicate; it does not return a list of the hosts. Examples of the commands are:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storageArray
defaultHostType;"
```

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storageArray
hostTypeTable;"
```

An example of defining a specific default host type is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set storagearray
defaultHostType=Microsoft2000;"
```

Setting the Storage Array Cache

The cache is high-speed memory designed to hold data that will either be written to the physical disks or read by the host. A controller has two memory areas used for intermediate storage of read and write data. The read cache contains data that has been read from the physical disks but not yet transferred to the host. The write cache contains data from the host but not yet written to the physical disks. The cache acts as a buffer so that data transfers between the host and physical disk does not need to be synchronized. In read caching, the data for a read operation from the host may already be in the cache from a previous operation, eliminating the need to access the physical disks. The data stays in the read cache until it is flushed. For write caching, a write operation stores data from the host in cache until it can be written to the physical disks.

The script command set provides two commands to define cache properties:

- `set storageArray`
- `set volume`

The `set storageArray` command enables you to change the cache block size, cache flush start value, and cache stop value. The general form of the command is:

```
set storageArray cacheBlockSize=cacheBlockSizeValue |  
cacheFlushStart=cacheFlushStartSize |  
cacheFlushStop=cacheFlushStopSize
```

NOTE You can enter one or all parameters on the command line when you run this command.

The cache block size value defines the size of the data block used by the controller in transferring data into or out of the cache. You can set the cache block size to either 4 KB or 16 KB. The value you use applies to the entire storage array and all volumes in the storage array. For redundant controller configurations, this includes all volumes owned by both controllers. Smaller cache block sizes should be used for systems that require transaction processing requests or I/O streams that are typically small and random. Larger cache block sizes should be used for large I/O, sequential, high bandwidth applications. The choice of block size will affect read/write performance. Large data transfers take longer in 4 KB block sizes than 16 KB sizes. The command for setting this parameter value would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set storageArray  
cacheBlockSize=16;"
```

To prevent data loss or corruption, the controller periodically writes cache data to disk (flushes the cache) when the amount of unwritten data in cache reaches a predefined level, called a start percentage, or when data has been in cache for a predetermined amount of time. The controller writes data to the physical disks until the amount of data in the cache drops to a stop percentage level. Using the `set storageArray` command, you can set

.....

the start and stop values as a percentages of the filled capacity of the cache. For example, you can specify that the controller start flushing the cache when it reaches 80 percent full and stop flushing the cache when it reaches 16 percent full. The command for setting these parameters this would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set storageArray
cacheFlushStart=80 cacheFlushStop=16;"
```

Low start and stop percentages provide for maximum data protection. For both low start and stop percentages; however, this increases the chance that data requested by a read command will not be in the cache. When the data is not in the cache, the cache hit percentage for writes and I/O requests decreases. Low start and stop values also increase the number of disk writes necessary to maintain the cache level, increasing system overhead and further decreasing performance.

The `set volume` command enables you to change settings for cache flush modifier, cache without batteries enabled, mirror cache enabled, read ahead multiplier, read cache enabled, and write cache enabled. You can use this command to set properties for all volumes or a specific volume in a volume group. The general form of the command is:

```
set (allVolumes | volume [volumeName] | volumes [volumeName1
... volumeNameN] | volume <wwn>) |
cacheFlushModifier=cacheFlushModifierValue |
cacheWithoutBatteryEnabled=(TRUE | FALSE) |
mirrorCacheEnabled=(TRUE | FALSE) | readCacheEnabled=(TRUE |
FALSE) | writeCacheEnabled=(TRUE | FALSE) |
readAheadMultiplier=integer-literal
```

NOTE You can enter one or all parameters on the command line when you run this command.

The `cacheFlushModifier` parameter defines the amount of time that data remains in cache before it is written to the physical disks. Table 3-5 lists the values for the cache flush modifier.

.....

Table 3-5 Cache Flush Modifier Values

Value	Meaning
Immediate	Data is flushed as soon as it is placed into the cache.
250	Data is flushed after 250 ms.
500	Data is flushed after 500 ms.
750	Data is flushed after 750 ms.
1	Data is flushed after 1 s.
1500	Data is flushed after 1500 ms.
2	Data is flushed after 2 s.
5	Data is flushed after 5 s.
10	Data is flushed after 10 s.
20	Data is flushed after 20 s.
60	Data is flushed after 60 s (1 min.).
120	Data is flushed after 120 s (2 min.).
300	Data is flushed after 300 s (5 min.).
1200	Data is flushed after 1200 s (20 min.).
3600	Data is flushed after 3600 s (1 hr.).
Infinite	Data in cache is not subject to any age or time constraints. The data will be flushed based on other criteria managed by the controller.

The command for setting this parameter value for all volumes in the storage array would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set allvolumes  
cacheFlushModifier=10;"
```

IMPORTANT Setting the cache flush modifier values above 10 seconds is not recommended. An exception is for testing purposes. After running any tests in which you have set the cache flush modifier values above 10 seconds, return the cache flush modifier value to 10 or fewer seconds.

The `cacheWithoutBatteryEnabled` parameter is a toggle that turns on or turns off the ability of a host to perform write caching without backup batteries in a controller. To enable write caching without batteries, set this parameter to **TRUE**. To disable write

.....

caching without batteries, set this parameter to **FALSE**. If you set this parameter to **TRUE** write caching continues, even when the controller batteries are completely discharged, not fully charged, or not present. If you do not have an uninterruptable power supply (UPS) and you enable this parameter, you can lose data if power to the storage array fails. The command for setting this parameter value would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volume  
[\"Engineering\"] cacheWithoutBatteryEnabled=false;"
```

The `mirrorCacheEnabled` parameter is a toggle that turns on or turns off write caching with mirroring. Write caching with mirroring enables cached data to be mirrored across redundant controllers that have the same cache size. Data written to the cache memory of one controller is also written to the cache memory of the second controller. If one controller fails, the second controller can complete all outstanding write operations. To use this option

- The controller pair must be an active/active pair.
- The controllers must have the same size cache.

To enable write caching with mirroring, set this parameter to **TRUE**. To disable write caching with mirroring, set this parameter to **FALSE**. The command for setting this parameter would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volume  
[\"Accounting\"] mirrorCacheEnabled=true;"
```

The `readCacheEnabled` parameter is a toggle that turns on or turns off the ability of the host to read data from the cache. Read caching enables read operations from the host to be stored in controller cache memory. If a host requests data that is not in the cache, the controller reads the needed data blocks from the disk and places them in the cache. Until the cache is flushed, all other requests for this data are fulfilled with cache data rather than from a physical disk read, increasing throughput. To enable read caching, set this parameter to **TRUE**. To disable read caching, set this parameter to **FALSE**. The command for setting this parameter would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volume  
[\"Balance 04\"] readCacheEnabled=true;"
```

The `writeCacheEnabled` parameter is a toggle that turns on or turns off the ability of the host to write data to the cache. Write caching enables write operations from the host to be stored in cache memory. The volume data in the cache is automatically written to the physical disks every 10 seconds. To enable write caching, set this parameter to **TRUE**. To disable write caching, set this parameter to **FALSE**. The command for setting this parameter would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set allVolumes  
writeCacheEnabled=true;"
```

The `readAheadMultiplier` parameter defines a *multiplier* value used to determine how many additional data blocks are read into cache. (You must set the

.....

readCacheEnabled parameter to **TRUE** before you set the read ahead multiplier value.) Valid read ahead multiplier values range from 1 to 65535. When you enter a multiplier value cache read ahead is enabled. When cache read ahead is enabled, the controllers can copy additional data blocks into the cache while the controller is reading and copying host requested data blocks from the physical disks into cache. Copying additional data blocks into the cache increases the chance that a future request for data can be fulfilled from the cache. Cache read ahead is important for applications that use sequential I/O, such as multimedia applications. Choosing a higher cache read ahead multiplier may increase the cache hit percentage; however, transfer time may also increase. To turn off cache read ahead, enter a read ahead multiplier value of 0. The command for setting this parameter value would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volumes
[\"Engineering 1\" \"Engineering 2\"]
readAheadMultiplier=500;"
```

Setting Modification Priority

Modification priority defines how much processing time is allocated for volume modification operations. Time allocated for volume modification operations affects system performance. Increases in volume modification priority can reduce read/write performance. Operations affected by modification priority include:

- Copyback
- Reconstruction
- Initialization
- Changing segment size
- Defragmentation of a volume group
- Adding free capacity to a volume group
- Changing the RAID level of a volume group

The lowest priority rate favors system performance, but the modification operation takes longer. The highest priority rate favors the modification operation, but the system performance may be degraded.

The `set volume` command enables you to define the modification priority for a volume. The general form of the command is:

```
set (allVolumes | volume [volumeName] | volumes [volumeName1
... volumeNameN] | volume <wwn> | accessVolume)
modificationPriority=(highest | high | medium | low |
lowest)
```

.....

An example of using this command to set the modification priority for volumes named “Engineering 1” and “Engineering 2” is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volumes
[\"Engineering 1\" \"Engineering 2\"]
modificationPriority=lowest;"
```

The modification rate is set to “lowest” so that system performance is not significantly reduced by modification operations.

Assigning Global Hot Spares

You can assign or unassign global hot spares using the `set drive` command. To use this command, you must identify the location of the physical disks by tray ID and slot ID and then set the `hotSpare` parameter to **TRUE** to enable the hot spare or **FALSE** to disable an existing hot spare. The general form of the command is:

```
set (drive [trayID,slotID] | drives [trayID1,slotID1 ...
trayIDn,slotIDn] hotSpare=(TRUE | FALSE)
```

An example of using this command to set hot spare physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set drives [1,2
1,3] hotSpare=true;"
```

Enter the tray ID and slot ID of each physical disk that you want to use. Enclose the list in square brackets, separate the tray ID and slot ID of a physical disk by a comma, and separate each tray ID and slot ID pair by a space.

Saving a Configuration to a File

After you have created a new configuration or if you want to copy an existing configuration for use on other storage arrays, you can save the configuration to a file using the `save storageArray configuration` command. Saving the configuration creates a script file you can run on the command line. The general form of the command is:

```
save storageArray configuration file="filename" [(allconfig
| globalSettings=(TRUE | FALSE)) |
volumeConfigAndSettings=(TRUE | FALSE) | hostTopology=(TRUE
| FALSE) | lunMappings=(TRUE | FALSE)]
```

IMPORTANT When you are writing information to a file, the script engine does not check to determine if the filename already exists. If you choose the name of a file that already exists, the script engine will write over the information in the file without warning.

.....

You can choose to save the entire configuration or specific configuration features. The command for setting this parameter value would look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "save storagearray
configuration file=\"c:\\folder\\storarrayconfig1.scr\";"
```

The file extension for a configuration file is “.scr.” The storage management software uses this extension when it creates the configuration file.

.....

Chapter 4

Using the Snapshot Premium Feature

This chapter describes the how the Snapshot premium feature works, lists the snapshot script commands, and explains how to use the commands to create snapshot volumes.

The Snapshot premium feature creates a snapshot volume that you can use as a backup of your data. A snapshot volume is a logical point-in-time image of a standard volume. Because it is not a physical copy, a snapshot volume is created more quickly than a physical copy and requires less physical disk space. Typically, you create a snapshot volume so that an application, such as a backup application, can access the snapshot volume and read the data while the base volume remains online and user accessible. You can also create several snapshot volumes of a base volume and write data to the snapshot volumes in order to perform testing and analysis.

Snapshot volumes include the following capabilities:

- Creating a complete image of the data on a base volume at a particular point-in-time
- Using only a small amount of disk space
- Providing for quick, frequent, non disruptive backups; or testing new versions of a database system without affecting real data
- Providing for snapshot volumes to be read, written, and copied
- Using the same availability characteristics of the base volume (such as, RAID protection and redundant path fail-over)
- Mapping the snapshot volume and making it accessible to any host on a storage area network (SAN). You can make snapshot data available to secondary hosts for read and write access by mapping the snapshot to the hosts
- Creating up to four snapshots per volume (the maximum number of snapshot volumes is one-half of the total number of volumes supported by the controller)
- Ability to increase the capacity of a snapshot volume

How Snapshot Works

Three components comprise a snapshot volume: the base volume, snapshot volume, and snapshot repository volume. Table 4-1 lists the components and provides brief descriptions of what they do.

Table 4-1 Snapshot Volume Components

Component	Description
Base volume	A standard volume from which the snapshot is created
Snapshot volume	A logical point-in-time image of a standard volume
Snapshot repository volume	A volume that contains snapshot metadata and copy-on-write data for a particular snapshot volume

Based on information you provide through the script commands, the storage management software creates an empty snapshot repository volume and defines the mapping from a base volume to the snapshot repository volume. The purpose of the snapshot repository volume is to hold “changed” data that a host writes to the base volume. When the snapshot repository volume is first created, it holds only the metadata about the snapshot volume with which it is associated.

NOTE When first creating a snapshot repository volume, briefly halt all write operations to the base volume so that a stable image of the base volume is available.

When the host writes to the base volume, the new data are also copied to the snapshot repository volume. (This is called *copy-on-write*.) A snapshot is constructed by combining the updated data in the repository volume with data in the base volume that have not been altered, creating a complete copy of the base volume at a specific point-in-time. The snapshot appears as a volume containing the original data at the time of creation, but it is actually an image that is the combination of the snapshot repository volume and the original base volume. The snapshot repository volume, which houses original data that have been changed, is the only additional physical disk space needed for the snapshot volume. The additional physical disk space is typically 10 to 20 percent of the base volume physical disk space and will vary depending on the amount of changes to the data. The longer a snapshot volume is active, the larger the repository needs to be. The default size of the repository volume is 20 percent of the base volume; however, you can set the repository volume size to other values.

.....

You can read, write, and copy a snapshot volume. Data written by a host to the snapshot are handled in the snapshot repository volume. When a write occurs to the base volume of a snapshot volume, the new data also overwrite the appropriate repository volume data.

Snapshot Commands

Table 4-2 lists the snapshot volume commands and brief descriptions of what the commands.

Table 4-2 Snapshot Volume Commands

Command	Description
<code>create snapshotVolume</code>	This command creates a snapshot volume.
<code>recreate snapshot</code>	This command starts a fresh copy-on-write operation using an existing snapshot volume.
<code>set (snapshotVolume)</code>	This command defines the properties for a snapshot volume and enables you to rename a snapshot volume.
<code>stop snapshot</code>	This command stops a copy-on-write operation.

Creating a Snapshot Volume

The `create snapshotVolume` command provides three methods for defining the physical disks for your snapshot repository volume:

- You define the each physical disks for the snapshot repository volume by tray ID and slot ID.
- You define a volume group in which the snapshot repository volume resides. You can optionally define the capacity of the repository volume.
- You define the number of drives, but not specific drives, for the repository volume.

When you use the `create snapshotVolume` command to create a snapshot volume, the minimum information you need to provide is the standard volume you want to use for the base volume. When you create a snapshot volume using minimum information, the storage management software provides default values for the other property parameters required for a completely defined snapshot volume.

Creating a Snapshot Volume with User Assigned Physical Disks

Creating a snapshot volume by assigning the physical disks provides flexibility in defining your configuration by enabling you to choose from the available physical disks in your storage array. When you choose the physical disks for your snapshot volume, you automatically create a new volume group. You can specify which physical disks to use and the RAID level for the new volume group. The general syntax for this command is:

```
create snapshotVolume baseVolume="baseVolumeName"  
[repositoryRAIDLevel=(1 | 3 | 5)  
(repositoryDrives=(trayID1,slotID1 ... trayIDn,slotIDn)  
userLabel="snapshotVolumeName"  
warningThresholdPercent=percentValue  
repositoryPercentOfBase=percentValue  
repositoryUserLabel="repositoryName"  
repositoryFullPolicy=(failBaseWrites | failSnapShot)]  
[trayLossProtect=(TRUE | FALSE)]
```

NOTE You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

An example of a command in which users assign the physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create  
snapshotVolume baseVolume=\"Mars_Spirit_4\  
repositoryRAIDLevel=5 repositoryDrives=(1,1 1,2 1,3 1,4  
1,5);"
```

.....

The command in this example creates a new snapshot of the base volume Mars_Spirit_4. The snapshot repository volume is comprised of five physical disks that form a new volume group. The new volume group will have a RAID level of five. This command also takes a snapshot of the base volume, starting the copy-on-write operation.

If you were to write this command in a script file, the command would look like:

```
create snapshotVolume baseVolume="Mars_Spirit_4"  
repositoryRAIDLevel=5 repositoryDrives=(1,1 1,2 1,3 1,4  
1,5);
```

A minimal version of this command might look like:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create  
snapshotVolume baseVolume=\"Mars_Spirit_4\";"
```

The command in this example creates a new snapshot for the base volume Mars_Spirit_4. The repository volume is created in the same volume group as the base volume, which means that the repository volume will have the same RAID level as the base volume. This command starts the copy-on-write operation.

The script file version of the command is:

```
create snapshotVolume baseVolume="Mars_Spirit_4";
```

Creating a Snapshot Volume with Software Assigned Physical Disks

With this version of the `create snapshotVolume` command, you choose an existing volume group in which to place the snapshot repository volume, and the storage management software determines which physical disks to use. You can also define how much space to assign to the repository volume. Because you are using an existing volume group, the RAID level for the snapshot volume defaults to the RAID level of the volume group in which you place it. You cannot define the RAID level for the snapshot volume. The general syntax for this command is:

```
create snapshotVolume baseVolume="baseVolumeName"  
[repositoryVolumeGroup=volumeGroupName  
repositoryFreeCapacityArea=freeCapacitySize  
userLabel="snapshotVolumeName"  
warningThresholdPercent=percentValue  
repositoryPercentOfBase=percentValue  
repositoryUserLabel="repositoryName"  
repositoryFullPolicy=(failBaseWrites | failSnapShot)]  
[trayLossProtect=(TRUE | FALSE)]
```

NOTE You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

An example of a command in which software assigns the physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create
snapshotVolume baseVolume=\"Mars_Spirit_4\"
repositoryVolumeGroup=2 repositoryFreeCapacityArea=250mb;"
```

The command in this example creates a new snapshot repository volume in volume group 2. The base volume is Mars_Spirit_4. The size of the snapshot repository is 4 GB. This command also takes a snapshot of the base volume, starting the copy-on-write operation.

When defining the capacity of a snapshot repository volume, a good working approximation for the size is 20 percent of the size of the base volume. In the previous example the size of the snapshot repository is set to 4 GB. The underlying assumption is that the base volume size is 20 GB (0.2 x 20 GB = 4 GB).

If you were to write this command in a script file the command would look like:

```
create snapshotVolume baseVolume="Mars_Spirit_4"
repositoryVolumeGroup=2 repositoryFreeCapacityArea=250mb;
```

Creating a Snapshot Volume by Specifying a Number or Physical Disks

With this version of the `create snapshotVolume` command you need to specify the number of physical disks and the RAID level that you want for the snapshot repository volume. This version of the `create snapshotVolume` command creates a new volume group. You must have physical disks in the storage array that are not assigned to a volume group for this command to work.

```
create snapshotVolume baseVolume="baseVolumeName"
[repositoryRAIDLevel=(1 | 3 | 5)
repositoryDriveCount=numberOfDrives driveType=(fibre | SATA
| PATA) userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryPercentOfBase=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failBaseWrites | failSnapShot)]
[trayLossProtect=(TRUE | FALSE)]
```

NOTE You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

.....

An example of a command in which users specify the number of physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create
snapshotVolume baseVolume=\"Mars_Spirit_4\"
repositoryRAIDLevel=5 repositoryDriveCount=3;"
```

The command in this example creates a new snapshot repository volume consisting of three physical disks. The three physical disks comprise a new volume group that has a RAID level of five. This command also takes a snapshot of the base volume, starting the copy-on-write operation.

If you were to write this command in a script file the command would look like:

```
create snapshotVolume baseVolume="Mars_Spirit_4"
repositoryRAIDLevel=5 repositoryDriveCount=3;
```

User Defined Parameters

Parameters for the `create snapshotVolume` command enable you to define the snapshot volume to suit the requirements of your storage array. Table 4-3 lists the parameters and descriptions of what the parameters do.

Table 4-3 Snapshot Volume Parameters

Parameter	Description
<code>driveType</code>	This parameter defines the type of drive you want to use for the snapshot repository volume. The choice is Fibre channel drive, SATA, or PATA. This parameter works only with the count based repository method of defining a snapshot volume.
<code>repositoryVolumeGroup</code>	This parameter defines the volume group in which you want to build the snapshot volume. The default is to build the snapshot repository volume in the same volume group as the base volume.
<code>repositoryFreeCapacityArea</code>	This parameter defines the amount of storage space you want to use for the snapshot repository volume. Free storage space is defined in units of bytes, KB, MB, or GB.

Parameter	Description
userLabel	This parameter defines the name you want to give to the snapshot volume. If you do not choose a name for the snapshot volume, the controllers create a default name using the base volume name. For example, if the base volume name is “Mars_Spirit_4” and does not have a snapshot volume, the default snapshot volume name is “Mars_Spirit_4-1.” If the base volume already has n-1 number of snapshot volumes, the default name is “Mars_Spirit_4-n.”
repositoryUserLabel	This parameter defines the name you want to give to the snapshot repository volume. If you do not choose a name for the snapshot repository volume, the controllers create a default name using the base volume name. For example, if the base volume name is “Mars_Spirit_4” and does not have an associated snapshot repository volume, the default snapshot repository volume name is “Mars_Spirit_4-R1.” If the base volume already has n-1 number of snapshot repository volumes, the default name is “Mars_Spirit_4-Rn.”
warningThresholdPercent	This parameter defines how full you will allow the snapshot repository volume to get before you receive a warning that the snapshot repository volume is nearing full. The warning value is a percentage of the total capacity of the snapshot repository volume. The default value is 50, which represents 50 percent of total capacity. (You can change this value later using the <code>set snapshotVolume</code> command.)
repositoryPercentOfBase	This parameter defines the size of the snapshot repository volume as a percentage of the base volume size. The default value is 20, which represents 20 percent of the base volume size.
repositoryFullPolicy	This parameter defines how you want snapshot processing to continue if the snapshot repository volume is full. You can choose to fail writes to the base volume (<code>failBaseWrites</code>) or fail writes to the snapshot volume (<code>failSnapshot</code>). The default value is <code>failSnapshot</code> .

.....

An example of the `create snapshotVolume` command that includes user defined parameters is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create
snapshotVolume baseVolume=\"Mars_Spirit_4\"
repositoryRAIDLevel=5 repositoryDriveCount=5 driveType=fibre
userLabel=\"Mars_Spirit_4_snap1\"
repositoryUserLabel=\"Mars_Spirit_4_rep1\"
warningThresholdPercent=75 repositoryPercentOfBase=40
repositoryFullPolicy=failSnapShot;"
```

If you were to write this command in a script file the command would look like:

```
create snapshotVolume baseVolume="Mars_Spirit_4"
repositoryRAIDLevel=5 repositoryDriveCount=5 driveType=fibre
userLabel="Mars_Spirit_4_snap1"
repositoryUserLabel="Mars_Spirit_4_rep1"
warningThresholdPercent=75 repositoryPercentOfBase=40
repositoryFullPolicy=failSnapShot;
```

NOTE In the previous examples, the names for the snapshot volume and repository volume are user defined. If you do not choose to create names for the snapshot or repository volumes the controllers will provide default names. (Refer to “Snapshot and Repository Volume Names” for an explanation about naming conventions.)

Snapshot and Repository Volume Names

You can use any combination of alphanumeric characters and the underscore (`_`) and dash (`-`) characters for the names. Names can have a maximum of 30 characters. You must enclose the name in double quotes. The character string cannot contain a new line. Make sure you use unique names; if you do not use unique names, the controller firmware will return an error.

One technique for naming the snapshot volume and the repository volume is to add a hyphenated suffix to the original base volume name. The suffix distinguishes between the snapshot volume and the repository volume. For example: If you have a base volume with a name of “Engineering Data” the snapshot volume can have a name of “Engineering Data-S1” and the repository volume can have a name of “Engineering Data-R1.”

If you do not choose a unique name for the either the snapshot volume or repository volume, the controllers create a default name using the base volume name. An example of the snapshot volume name the controllers might create is: if the base volume name is “aaa” and does not have a snapshot volume, the default snapshot volume name is “aaa-1;” if the base volume already has n-1 number of snapshot volumes, the default name is “aaa-

.....

n.” An example of the repository volume name the controller might create is: if the base volume name is “aaa” and does not have a repository volume, the default repository volume name is “aaa-R1;” if the base volume already has n-1 number of repository volumes, the default name is “aaa-Rn.”

In the examples from the previous section, the user defined snapshot volume name was “Mars_Spirit_4_snap1” and the user defined repository volume name was “Mars_Spirit_4_rep1”. The controller provided default name for the snapshot volume would be “Mars_Spirit_4-1”. The controller provided default name for the repository volume would be “Mars_Spirit_4-R1”.

Changing Snapshot Volume Settings

The `set (snapshot) volume` command enables you to change the property settings for a snapshot volume. Using this command you can change the

- Snapshot volume name
- Warning threshold percent
- Repository full policy

An example of changing a snapshot volume name is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volume
[\"Mars_Spirit_4-1\"] userLabel=\"Mars_Odyssey_3-2\";"
```

If you were to write this command in a script file, the command would look like:

```
set volume [\"Mars_Spirit_4-1\"] userLabel=\"Mars_Odyssey_3-
2\";
```

When you change the warning threshold percent and repository full policy, you can apply the changes to one or several snapshot volumes with this command. An example of using the `set (snapshot) volume` command to change these properties on more than one snapshot volume is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volumes
[\"Mars_Spirit_4-1\" \"Mars_Spirit_4-2\" \"Mars_Spirit_4-
3\"] warningThresholdPercent=50
repositoryFullPolicy=failBaseWrites;"
```

If you were to write this command in a script file, the command would look like:

```
set volumes ["Mars_Spirit_4-1" "Mars_Spirit_4-2"
"Mars_Spirit_4-3"] warningThresholdPercent=50
repositoryFullPolicy=failBaseWrites;
```

Stopping, Restarting, and Deleting a Snapshot Volume

When you create a snapshot volume, copy-on-write immediately starts running. As long as a snapshot volume is enabled, storage array performance is impacted by the copy-on-write operations to the associated snapshot repository volume. If you no longer want copy-on-write operations to run, you can use the `stop snapshot volumes` command to stop the copy-on-write operations. When you stop a snapshot volume, the snapshot volume and the repository volume are still defined for the base volume; only copy-on-write has stopped. An example of stopping a snapshot volume is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "stop snapshot volumes [\"Mars_Spirit_4-2\" \"Mars_Spirit_4-3\"];"
```

If you were to write this command in a script file the command would look like:

```
stop snapshot volumes ["Mars_Spirit_4-2" "Mars_Spirit_4-3"];
```

When you stop the copy-on-write operations for a specific snapshot volume, only that snapshot volume is disabled. All other snapshot volumes remain in operation.

When you want to restart a copy-on-write operation, use the `recreate snapshot volume` command. This command starts a fresh copy-on-write operation using an existing snapshot volume. When you restart a snapshot volume

- The snapshot volume must have either an Optimal or a Disabled state.
- All copy-on-write data previously on the snapshot repository volume is deleted.
- Snapshot volume and snapshot repository volume parameters remain the same as the previously disabled snapshot volume and snapshot repository volume. You can also change the `userLabel`, `warningThresholdPercent`, and `repositoryFullPolicy` parameters when you restart the snapshot volume.
- The original names for the snapshot repository volume is retained.

An example of restarting a snapshot volume is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "recreate snapshot volumes [\"Mars_Spirit_4-2\" \"Mars_Spirit_4-3\"];"
```

If you were to write this command in a script file the command would look like:

```
recreate snapshot volumes ["Mars_Spirit_4-2" "Mars_Spirit_4-3"];
```

If you do not intend to use a snapshot volume again, you can delete the snapshot volume using the `delete volume` command. When you delete a snapshot volume, the associated snapshot repository volume is also deleted.

Chapter 5

Using the Remote Volume Mirroring Premium Feature

This chapter describes how Remote Volume Mirroring works, lists the Remote Volume Mirroring script commands, and explains how to use the commands to create and run remote volume pairs. Additionally, this chapter describes considerations for using other premium features with the Remote Volume Mirror premium feature.

The Remote Volume Mirroring premium feature provides for online, real-time replication of data between storage arrays over a remote distance. In the event of a disaster or catastrophic failure on one storage array, you can promote the second storage array to take over responsibility for computing services. Remote Volume Mirroring is designed for extended storage environments in which the storage arrays used for Remote Volume Mirroring are maintained at separate sites. Volumes on one storage array are mirrored to volumes on another storage array across a fabric SAN. Data transfers can be synchronous or asynchronous. You choose the method when setting up the Remote Volume Mirror pair. The data transfers occur at Fibre Channel speeds to maintain data on the different storage arrays. Because Remote Volume Mirroring is storage-based, it does not require any server or application overhead.

You can use Remote Volume Mirroring for:

- Disaster recovery
Remote Volume Mirroring enables you to replicate data from one site to another site, providing an exact mirror duplicate at the remote (secondary) site. If the primary site fails, mirrored data at the remote site can be used for fail over and recovery. You can then shift storage operations to the remote site for continued operation of all services normally provided by the primary site.
- Data vaulting and data availability
Remote Volume Mirroring enables you to send data off site where it can be protected. You can then use the off site copy for testing or to act as a source for a full backup to avoid interrupting operations at the primary site.
- Two-way data protection
Remote Volume Mirroring provides the ability to have two storage arrays back up each other by mirroring critical volumes on each storage array to volumes on the other storage array. This enables each storage array to recover data from the other system in the event of any service interruptions.

How Remote Volume Mirroring Works

When you create a Remote Volume Mirror pair, the mirrored pair consists of a *primary volume* on a local storage array and a *secondary volume* on a storage array at another site. A standard volume may only be included in one mirrored volume pair. You can define up to 32 mirrors on a 2882 storage array and up to 64 mirrors on 4884 and 5884 storage arrays.

The primary volume is the volume that accepts host I/O activity and stores application data. When the mirror relationship is first created, data from the primary volume is copied in its entirety to the secondary volume. This process is known as a *full synchronization* and is directed by the controller owner of the primary volume. During a full synchronization, the primary volume remains fully accessible for all normal I/O operations.

The controller owner of the primary volume initiates remote writes to the secondary volume to keep the data on the two volumes synchronized.

The secondary volume maintains a mirror (or copy) of the data on its associated primary volume. The controller owner of the secondary volume receives remote writes from the primary volume controller owner, but will not accept host write requests. Hosts are able to read from secondary volume, which appears as read-only.

In the event of a disaster or catastrophic failure at the primary site, you can perform a role reversal to promote the secondary volume to a primary role. Hosts will then be able to read and write to the newly promoted volume and business operations can continue.

Mirror Repository Volumes

A mirror repository volume is a special volume in the storage array created as a resource for the controller owner of the primary volume in a Remote Volume Mirror. The controller stores mirroring information on this volume, including information about remote writes that are not yet complete. The controller can use this information to recover from controller resets and accidentally powering-down the storage arrays.

When you activate the Remote Volume Mirroring premium feature on the storage array, you create two mirror repository volumes, one for each controller in the storage array. An individual mirror repository volume is not needed for each Remote Volume Mirror.

When you create the mirror repository volumes, you specify the location of the volumes. You can either use existing free capacity or you can create a volume group for the volumes from unconfigured capacity and then specify the RAID level.

Because of the critical nature of the data being stored, the RAID level of mirror repository volumes must not be RAID 0 (for data striping). The required size of each volume is 128 MB, or 256 MB total for both mirror repository volumes of a dual controller array. In previous versions of the Remote Volume Mirroring feature, the mirror repository volumes required less disk space, and need to be upgraded in order to utilize the maximum amount of mirror relationships.

Mirror Relationships

Before creating a mirror relationship, you must enable the Remote Volume Mirroring premium feature on both the primary and secondary storage arrays. You must also create a secondary volume on the secondary site if one does not already exist. The secondary volume must be a standard volume of equal or greater capacity than the associated primary volume.

When secondary volumes are available, you can establish a mirror relationship in the storage management software by identifying the primary volume and the storage array containing the secondary volume.

When you first create the mirror relationship, a full synchronization automatically occurs, with data from the primary volume copied in its entirety to the secondary volume.

Data Replication

Data replication between the primary volume and the secondary volume is managed by the controllers and is transparent to host machines and applications. This section describes how data is replicated between storage arrays participating in Remote Volume Mirroring and the actions taken by the controller owner of the primary volume if a link interruption occurs between storage arrays.

Write Modes

When the controller owner of the primary volume receives a write request from a host, the controller first logs information about the write to a mirror repository volume, and then writes the data to the primary volume. The controller then initiates a remote write operation to copy the affected data blocks to the secondary volume at the secondary storage array.

The Remote Volume Mirror feature provides two write mode options that affect when the I/O completion indication is sent back to the host: synchronous and asynchronous.

Synchronous

Synchronous write mode provides the highest level security for full data recovery from the secondary storage array in the event of a disaster. Synchronous write mode does, however, reduce host I/O performance. When this write mode is selected, host write requests are written to the primary volume and then copied to the secondary volume. After the host write request has been written to the primary volume and the data has been successfully copied to the secondary volume, the controller removes the log record on the mirror repository volume and sends an I/O completion indication back to the host system. Synchronous write mode is selected by default and is the recommended write mode.

Asynchronous

Asynchronous write mode offers faster host I/O performance but does not guarantee that a copy operation has successfully completed before processing the next write request. When you use asynchronous write mode, host write requests are written to the primary volume. The controller then sends an “I/O complete” indication back to the host system, without acknowledging that the data has been successfully copied to the secondary (remote) storage array.

When using asynchronous write mode, write requests are not guaranteed to be completed in the same order on the secondary volume as they are on the primary volume. If the order of write requests is not retained, data on the secondary volume may become inconsistent with the data on the primary volume, and could jeopardize any attempt to recover data if a disaster occurs on the primary storage array.

Write Consistency Groups

When multiple mirror relationships exist on a single storage array and have been configured to use asynchronous write mode and to preserve consistent write order, they are considered to be an interdependent group called a *write consistency group*. The data on the secondary, remote storage array cannot be considered fully synchronized until all Remote Volume Mirrors in the write consistency group are synchronized.

If one mirror relationship in a write consistency group becomes unsynchronized, all of the mirror relationships in the write consistency group will become unsynchronized, and any write activity to the remote, secondary storage arrays will be prevented to protect the consistency of the remote data set.

Link Interruptions or Secondary Volume Errors

When processing write requests, the primary controller may be able to write to the primary volume, but a link interruption may prevent communication with the remote (secondary) controller.

In this case, the remote write operation cannot be completed to the secondary volume, and the primary and secondary volumes are no longer correctly mirrored. The primary controller transitions the mirrored pair into an unsynchronized state and sends an I/O completion to the primary host. The primary host can continue to write to the primary volume, but remote writes will not take place.

When communication is restored between the controller owner of the primary volume and the controller owner of the secondary volume, a resynchronization will take place either automatically or will need to be started manually, depending on which write mode you chose when setting up the mirror relationship. During the resynchronization, only the blocks of data that have changed on the primary volume during the link interruption are copied to the secondary volume. After the resynchronization begins, the mirrored pair will transition from an unsynchronized status to a synchronization in progress status.

.....

The primary controller will also mark the mirrored pair as unsynchronized when a volume error on the secondary side prevents the remote write from completing. For example, an offline or a failed secondary volume can cause the Remote Volume Mirror to become unsynchronized. When the volume error is corrected (the secondary volume is placed online or recovered to an optimal status), then synchronization is required, and the mirrored pair transitions to a synchronization in progress status.

Resynchronization

Data replication between the primary volume and secondary volume in a mirror relationship is managed by the controllers and is transparent to host machines and applications. When the controller owner of the primary volume receives a write request from a host, the controller first logs information about the write to a mirror repository volume, and then writes the data to the primary volume. The controller then initiates a write operation to copy the affected data to the secondary volume on the remote storage array.

If a link interruption or volume error prevents communication with the secondary storage array, the controller owner of the primary volume transitions the mirrored pair into an unsynchronized status and sends an I/O completion to the host sending the write request. The host can continue to issue write requests to the primary volume, but remote writes to the secondary volume will not take place.

When connectivity is restored between the controller owner of the primary volume and the controller owner of the secondary volume, the volumes must be resynchronized by copying the blocks of data changed during the interruption to the secondary volume. Only the blocks of data that have changed on the primary volume during the link interruption will be copied to the secondary volume.

CAUTION Potential data corruption. Any communication disruptions between the primary and secondary storage array while resynchronization is underway could result in a mix of new and old data on the secondary volume. This would render the data unusable in a disaster recovery situation.

Remote Volume Mirror Commands

The Remote Volume Mirror commands and brief descriptions of what the commands do are listed in the following table:

.....

Table 5-1 Remote Volume Mirror Commands

Command	Description
<code>activate storageArray feature=remoteMirror</code>	This command creates the mirror repository volume and activates the Remote Volume Mirror feature. When you use this command you can define the mirror repository volume in one of three ways: by specific physical disks, by a specific volume group, by a number of physical disks.
<code>create remoteMirror</code>	This command creates both the primary and secondary volumes for a Remote Volume Mirror. This command also sets the write mode (synchronous or asynchronous) and synchronization priority.
<code>deactivate storageArray feature=remoteMirror</code>	This command deactivates the Remote Volume Mirror feature and tears down the repository volume. The host port dedicated to the Remote Volume Mirror is made available for host I/O.
<code>diagnose remoteMirror</code>	This command tests the connection between the specified primary volumes and mirror volumes on a storage array with the Remote Volume Mirror feature installed.
<code>recreate storageArray mirrorRepository</code>	This command creates a new Remote Volume Mirror repository volume using the parameters defined for a previous Remote Volume Mirror repository volume. When you use this command you can define the mirror repository volume in one of three ways: by specific physical disks, by a specific volume group, or by a number of physical disks.
<code>remove remoteMirror</code>	This command removes the mirror relationship between the primary and secondary volume.
<code>resume remoteMirror</code>	This command resumes a suspended Remote Volume Mirror operation.

Command	Description
<code>set remoteMirror</code>	This command defines the properties for a Remote Volume Mirror pair.
<code>show remoteMirror candidates</code>	This command returns information about the candidate volumes on the remote storage array that you can use as secondary volumes for a primary volume.
<code>show remoteMirror synchronizationProgress</code>	This command returns the progress of data synchronization between the primary volume and secondary volume. This command displays the progress as a percentage of data synchronization completed.
<code>start remoteMirror synchronize</code>	This command starts Remote Volume Mirror synchronization.
<code>suspend remoteMirror</code>	This command suspends a Remote Volume Mirror operation.

Creating a Remote Volume Mirror Pair

Before you create any mirror relationships, volumes must exist at both the primary and secondary sites. The volume residing on the local storage array is the primary volume. Similarly, the volume residing on the remote storage array is the secondary volume. If neither the primary or secondary volumes exist, you will need to create these volumes. When you create the secondary volume you must take the following into consideration:

- The secondary volume must be of equal or greater size than the primary volume.
- The RAID level of the secondary volume does not have to be the same as the primary volume.

The general steps for creating a Remote Volume Mirror pair are

1. Enable the Remote Volume Mirror feature.
2. Activate the Remote Volume Mirror feature.
3. Determine candidates for a Remote Volume Mirror pair.
4. Create the Remote Volume Mirror relationship.

Performance Considerations

The following general performance considerations should be noted when creating mirror relationships.

- The controller owner of a primary volume performs a full synchronization in the background while processing local I/O writes to the primary volume and associated remote writes to the secondary volume. Because the full synchronization diverts controller processing resources from I/O writes, it can have a performance impact to the host application.

To reduce the performance impact, you can set the synchronization priority level to determine how the controller owner will prioritize the full synchronization relative to other I/O activity. The basic guidelines for setting the synchronization priority level are

- A full synchronization at the lowest synchronization priority level will take approximately eight times as long as a full synchronization at the highest synchronization priority level.
- A full synchronization at the low synchronization priority level will take approximately six times as long as a full synchronization at the highest synchronization priority level.

-
- A full synchronization at the medium synchronization priority level will take approximately three and a half times as long as a full synchronization at the highest synchronization priority level.
 - A full synchronization at the high synchronization priority level will take approximately twice as long as a full synchronization at the highest synchronization priority level.
 - When the mirrored volume pair is in a Synchronization in Progress state, all host write data is copied to the remote system. Both controller I/O bandwidth and I/O latency can affect host write performance. Host read performance is not affected by the mirroring relationship.
 - The time that it takes for data to be copied from the primary volume to the secondary volume may impact overall performance and is primarily because of the delay and system resource required for copying data to the Remote Volume Mirror. Some delay may also occur because of the limit to the number of simultaneous writes.

Enabling the Remote Volume Mirror Feature

The first step in creating a Remote Volume Mirror is to make sure the feature is enabled on both storage arrays. Because Remote Volume Mirror is a premium feature, you will need a feature key to enable the feature. The command for enabling the feature key file is:

```
enable storageArray feature file="filename"
```

where the `filename` parameter is the complete file path and filename of a valid feature key file. Enclose the file path and filename in double quotes (“ ”). Valid filenames for feature key files end with a `.key` extension.

Activating the Remote Volume Mirror Feature

Activating the Remote Volume Mirror feature prepares the storage arrays to create and configure mirror relationships. After you activate the feature the secondary ports for each controller are reserved and dedicated to Remote Volume Mirror usage. In addition, a mirror repository volume is automatically created for each controller in the storage array. As part of the activation process you will have the option to decide where the mirror repository volumes will reside, free capacity on an existing volume group or in a newly created volume group, and the RAID level for the repository volumes.

The free capacity you select for the mirror repository volume must have a total of 256 MB capacity available. Two mirror repository volumes are created on this capacity, one for each controller. If you enter a value for the repository storage space that is too small for the mirror repository volumes, the firmware will return an error message giving the amount of space needed for the mirror repository volumes. The command will not attempt to activate the Remote Volume Mirror. You can re-enter the command using the value from the error message for the repository storage space value.

.....

The RAID level you choose for the repository volume has the following constraints:

- RAID 0: You cannot use RAID 0.
- RAID 1: The number of drives must be an even number. If you select an odd number of drives, controller firmware returns an error.
- RAID 3 or 5: You must have a minimum of three physical disks in the volume group.

The general form for activating the Remote Volume Mirror feature is:

```
activate storageArray feature=remoteMirror
```

The `activate storageArray feature=remoteMirror` command provides three methods for defining the physical disks for your repository volume:

- You define each physical disk for the repository volume by tray ID and slot ID.
- You define a volume group in which the repository volume resides. You can optionally define the capacity of the repository volume.
- You define the number of drives, but not specific drives, for the repository volume.

Activating the Remote Volume Mirror Feature with User Assigned Physical Disks

Activating the Remote Volume Mirror feature by assigning the physical disks provides flexibility in defining your configuration by enabling you to choose from the available physical disks in your storage array. Choosing the physical disks for your Remote Volume Mirror automatically creates a new volume group. You can specify which physical disks to use and the RAID level for the new volume group. The general syntax for this command is:

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5)
repositoryDrives=(trayID1,slotID1 ... trayIDn,slotIDn)
[trayLossProtect=(TRUE | FALSE)]
```

An example of a command in which you assign the physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "activate
storageArray feature=remoteMirror repositoryRAIDLevel=5
repositoryDrives=(1,1 1,2 1,3 1,4 1,5);"
```

The command in this example creates a new mirror repository volume comprised of five physical disks that form a new volume group. The new volume group will have a RAID level of five.

.....

If you were to write this command in a script file, the command would look similar to the following example:

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=5 repositoryDrives=(1,1 1,2 1,3 1,4
1,5);
```

Activating the Remote Volume Mirror Feature with Software-Assigned Physical Disks

With this version of the `activate storageArray feature=remoteMirror` command, you choose an existing volume group in which to place the mirror repository volume, the storage management software then determines which physical disks to use. You can also define how much space to assign to the mirror repository volume. Because you are using an existing volume group, the RAID level for the mirror repository volume defaults to the RAID level of the volume group in which you place it. You cannot define the RAID level for the mirror repository volume. The general syntax for this command is:

```
activate storageArray feature=remoteMirror
repositoryVolumeGroup=volumeGroupName
[repositoryFreeCapacityArea=freeCapacityValue]
[trayLossProtect=(TRUE | FALSE)]
```

An example of a command in which software assigns the physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "activate
storageArray feature=remoteMirror repositoryVolumeGroup=2
repositoryFreeCapacityArea=250mb;"
```

The command in this example creates a new mirror repository volume in volume group 2. The size of the mirror repository volume is 250 MB.

If you were to write this command in a script file, the command would look similar to the following example:

```
activate storageArray feature=remoteMirror
repositoryVolumeGroup=2 repositoryFreeCapacityArea=250mb;
```

Activating the Remote Volume Mirror Feature by Specifying a Number or Physical Disks

With this version of the `activate storageArray feature=remoteMirror` command, you need to specify the number of physical disks and the RAID level that you want for the mirror repository volume. This version of the `activate storageArray feature=remoteMirror` command creates a new volume group. For this command to work, you must have physical disks in the storage array that are not assigned to a volume group.

.....

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5)
repositoryDriveCount=numberOfDrives [driveType=(fibre | SATA
| PATA)] [trayLossProtect=(TRUE | FALSE)]
```

An example of a command in which you specify the number of physical disks is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "activate
storageArray feature=remoteMirror repositoryRAIDLevel=5
repositoryDriveCount=5 driveType=fibre;"
```

The command in this example creates a new mirror repository volume using five software-selected physical disks for the repository volume. The mirror repository volume has a RAID level of five. The type of drive for the mirror repository volume is Fibre Channel.

If you were to write this command in a script file, the command would look similar to the following example:

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=5 repositoryDriveCount=5
driveType=fibre;
```

Determining Candidates for a Remote Volume Mirror Pair

All volumes and physical disks on the remote storage array may not be available for use as secondary volumes. To determine which volumes on a remote storage array you can use as candidates for secondary volumes, use the `show remoteMirror candidates` command. This command returns a list of the volumes you can use when creating a Remote Volume Mirror. The general form of the command is:

```
show remoteMirror candidates primary="volumeName"
remoteStorageArray="storageArrayName"
```

where `volumeName` is the name of the volume you want to use for the primary volume and `storageArrayName` is the remote storage array that contains possible candidates for the secondary volume. Enclose both the volume name and the storage array name in double quotes ("").

Creating a Remote Volume Mirror Pair

When you create a new Remote Volume Mirror, you must define which volumes you want to use for the primary (local) and secondary (remote) volumes. You define the primary volume by the name of the volume. You define the secondary volume by name with either the name or WWN of the storage array the secondary volume is on. The primary volume name, secondary volume name, and remote storage array name (or WWN) are the minimum information you need to provide. Using this command, you can also define synchronization priority, write order, and write mode.

.....

The general form of the command is:

```
create remoteMirror primary="primaryVolumeName"
secondary="secondaryVolumeName"
(remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwn="wwn") [remotePassword=password
syncPriority=(highest | high | medium | low | lowest)
writeOrder=(preserved | notPreserved) writeMode=(synchronous
| asynchronous)]
```

NOTE You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

An example of the `create remoteMirror` command is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create remoteMirror
primary=\"Jan_04_Account\" secondary=\"Jan_04_Account_B\"
remotestoragearrayname=\"Tabor\" remotepassword=\"jdw2ga05\"
syncpriority=highest writemode=synchronous;"
```

The command in this example creates a Remote Volume Mirror in which the primary volume is named “Jan_04_Account” on the local storage array and the secondary volume is named “Jan_04_Account_B” on the remote storage array that is named “Tabor.” The names used in this example are similar, but that is not a requirement for the volume names in a Remote Volume Mirror pair. In this example, the remote storage array has a password that you must enter when making any change to the storage array configuration. Creating a Remote Volume Mirror pair is a significant change to a storage array configuration. Setting the write mode to “synchronous” and the synchronization priority to “highest” means that host write requests are written to the primary volume and then immediately copied to the secondary volume. This helps ensure that the data on the secondary volume is as accurate a copy of the data on the primary as possible. The highest synchronization priority does, however, use more system overhead, which can reduce system performance.

If you were to write this command in a script file, the command would look similar to the following example:

```
create remoteMirror primary="Jan_04_Account"
secondary="Jan_04_Account_B" remotestoragearrayname="Tabor"
remotepassword="jdw2ga05" syncpriority=highest
writemode=synchronous;"
```

After you have created a Remote Volume Mirror, you can see the progress of data synchronization between the primary and secondary volumes by running the `show remoteMirror synchronizationProgress` command. This command displays the progress as a percentage of data synchronization completed.

Changing Remote Volume Mirror Settings

The `set remoteMirror` command enables you to change the property settings for a Remote Volume Mirror pair. Using this command, you can change the following property settings:

- Volume role (either primary or secondary)
- Synchronization priority
- Write order
- Write mode

You can apply the changes to one or several Remote Volume Mirror pairs with this command. Use the primary volume name to identify the mirror pairs for which you are changing the properties.

An example of using the `set remoteMirror` command to change these parameters is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set remoteMirror
localVolume [\"Jan_04_Account\"] syncpriority=medium
writeorder=notpreserved writemode=asynchronous;"
```

If you were to write this command in a script file the command would look similar to the following example:

```
set remoteMirror localVolume ["Jan_04_Account"]
syncpriority=medium writeorder=notpreserved
writemode=asynchronous;
```

Suspending and Resuming a Mirror Relationship

Use the `suspend remoteMirror` command to stop data transfer between a primary volume and secondary volume in a mirror relationship without disabling the mirror relationship. Suspending a mirror relationship enables you to control when the data on the primary volume and secondary volume are synchronized. Suspending a mirror relationship helps reduce any performance impact to the host application that may occur while any changed data on the primary volume is copied to the secondary volume. Suspending a mirror relationship is particularly useful when you want to run a back up of the data on the secondary volume.

When a mirror relationship is in a suspended state, the primary volume does not make any attempt to contact the secondary volume. Any writes to the primary volume are persistently logged in the mirror repository volumes. After the mirror relationship resumes, any data that is written to the primary volume is automatically written to the secondary volume. Only the modified data blocks on the primary volume are written to the secondary volume. Full synchronization is not required.

IMPORTANT If you suspend a Remote Volume Mirror that is part of a write consistency group, you will suspend all mirrored pairs in the write consistency group. You can, however, then resume mirror operations for any of the individual mirror pairs in the write consistency group.

An example of using the `suspend remoteMirror` command is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "suspend
remotemirror volume ["Jan_04_Account"]
writeconsistency=false;"
```

The `writeConsistency` parameter defines whether the volumes identified in this command are in a write-consistency group or are separate. For the volumes in a write-consistency group, set this parameter to `TRUE`. For the volumes that are not in a write consistency group, set this parameter to `FALSE`.

If you were to write this command in a script file, the command would look similar to the following example:

```
suspend remotemirror volume ["Jan_04_Account"]
writeconsistency=false;
```

The mirror relationship remains suspended until you use the `resume remoteMirror` command to restart synchronization activities. The `resume remoteMirror` command restarts data transfers between a primary volume and secondary volume in a mirror relationship after the mirror has been suspended or unsynchronized.

.....

An example of using the `resume remoteMirror` command is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "resume remotemirror  
volume [\"Jan_04_Account\"] writeconsistency=false;"
```

The `writeConsistency` parameter defines whether the volumes identified in this command are in a write-consistency group or are separate. For the volumes in the same write-consistency group, set this parameter to `TRUE`. For the volumes that are separate, set this parameter to `FALSE`.

If you were to write this command in a script file, the command would look similar to the following example:

```
resume remotemirror volume [\"Jan_04_Account\"]  
writeconsistency=false;
```

Removing a Mirror Relationship

Use the `remove remoteMirror` command to remove the link between a primary volume and secondary volume. (Removing a mirror relationship is similar to deleting a mirror relationship.) Removing the link between a primary and secondary volume does not affect any of the existing data on either volume. The link between the volumes is removed, but the primary volume still continues normal I/O operations. You can later establish the mirror relationship between the two volumes and resume normal mirror operations. You can remove the mirror relationship for one or several Remote Volume Mirror pairs with this command.

An example of using the `remove remoteMirror` command is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "remove remotemirror  
localVolume [\"Jan_04_Account\"];"
```

When you run this command, use the name of the primary volume of the mirror pair.

If you were to write this command in a script file, the command would look similar to the following example:

```
remove remotemirror localVolume ["Jan_04_Account"];
```

To re-establish the link between a primary volume and secondary volume, use the `create remoteMirror` command.

Deleting a Primary or Secondary Volume

Use the `delete volume` command to remove a primary or secondary volume from a storage array. Deleting a volume in a mirror relationship removes the mirror relationship and completely deletes the volume from the storage array. The mirror relationship cannot be redefined until you create a new volume or choose an alternate volume to replace the deleted volume.

IMPORTANT Deleting a primary or secondary volume permanently removes the data from the storage array.

Disabling the Remote Volume Mirror Feature

The purpose of disabling the Remote Volume Mirror feature is to prevent new mirror relationship from being created. When you disable the Remote Volume Mirror feature, the feature is in a disabled/active state. In this state, you can maintain and manage previously existing mirror relationships; however, you cannot create new relationships. To disable the Remote Volume Mirror feature, run the following command:

```
disable storageArray feature=remoteMirror
```

Deactivating the Remote Volume Mirror Feature

If you no longer require the Remote Volume Mirror feature and you have removed all mirror relationships, you can deactivate the feature. Deactivating the feature reestablishes normal use of dedicated ports on both storage arrays and deletes both mirror repository volumes. To deactivate the Remote Volume Mirror feature, run the following command:

```
deactivate storageArray feature=remoteMirror
```

Interaction with Other Features

You can run the Remote Volume Mirror feature while running the following premium features:

- Storage Partitioning
- Snapshot Volumes
- Volume Copy

When you are running the Remote Volume Mirror feature with other premium features, you will need to take the requirements of other premium features into consideration to help ensure you set up a stable storage array configuration.

In addition to the premium features, you can also run the Remote Volume Mirror feature while running Dynamic Volume Expansion.

Storage Partitioning

Storage partitioning is a premium feature that enables hosts to share access to volumes in a storage array. You create a storage partition when you define any of the following storage array logical components:

- A host
- A host group
- A volume-to-LUN mapping

The volume to LUN mapping enables you to define which host group or host has access to a particular volume in the storage array.

When you are creating storage partitions, define the partitions after you have created the primary and secondary volumes in a Remote Volume Mirror configuration. The storage partition definitions for the primary and secondary storage arrays are independent of each other. If these definitions are put in place while the volume is in a secondary role, it will reduce the administrative effort associated with the site recovery if it becomes necessary to promote the volume to a primary role.

Snapshot Volumes

A snapshot volume is a point-in-time image of a volume and is typically created so that an application, such as a backup application, can access the snapshot volume and read the data while the base volume remains online and accessible to hosts.

The volume for which the point-in-time image is created is known as the base volume and must be a standard volume in the storage array. The snapshot repository volume stores information about all data altered since the snapshot was created.

.....

In this version of the storage management software, you can create snapshot volumes based on the primary volume or secondary volume of a Remote Volume Mirror.

Volume Copy

The Volume Copy premium feature copies data from one volume (the source volume) to another volume (the target volume) within a single storage array. You can use this feature to copy data from volume groups that use smaller capacity drives to volume groups that use larger capacity drives, to back up data, or to restore snapshot volume data to the base volume.

You can use a primary volume in a Remote Volume Mirror as a source volume or a target volume in a volume copy. You cannot use a secondary volume as a source or target volume. If you initiate a role reversal during a copy in progress, the copy will fail and cannot be restarted.

Dynamic Volume Expansion

A Dynamic Volume Expansion (DVE) is a modification operation used to increase the capacity of a standard volume or a snapshot repository volume. The increase in capacity is achieved by using the free capacity available in the volume group of the standard or snapshot repository volume.

This modification operation is considered to be “dynamic” because you have the ability to continually access data on volume groups, volumes, and physical disks throughout the entire operation.

A DVE operation can be performed on a primary or secondary volume of a mirror relationship; however, even though the storage management software indicates that the volume has increased capacity, its usable capacity is the size of the smaller of the primary or secondary volumes.

You cannot perform a DVE operation on a mirror repository volume.

Chapter 6

Using the Volume Copy Premium Feature

This chapter describes how the Volume Copy premium feature works, lists the script commands for volume copy, and explains how to use the commands to create and run volume copy. Additionally, this chapter describes considerations for using other premium features with the Volume Copy premium feature.

The Volume Copy premium feature enables you to copy data from one volume (the source) to another volume (the target) in a single storage array. You can use this feature to back up data, copy data from volume groups that use smaller capacity drives to volume groups using greater capacity drives, or restore snapshot volume data to the associated base volume.

How Volume Copy Works

When you create a volume copy, you create a copy pair that consists of a source volume and a target volume. Both the source volume and target volume are located on the same storage array. During a volume copy, the controllers manage copying the data from the source volume to the target volume. The volume copy is transparent to the host machines and applications, except that users cannot write to the source volume during a volume copy.

While a volume copy is In Progress, the same controller must own both the source volume and target volume. If one controller does not own both the source volume and target volume before creating the volume copy, ownership of the target volume is automatically transferred to the controller that owns the source volume. When the volume copy is finished, or stopped, ownership of the target volume is restored to its preferred controller. If ownership of the source volume changes while a volume copy is running, ownership of the target volume also changes.

Source Volume

The source volume is the volume that accepts host I/O and stores data. When you start a volume copy, data from the source volume is copied in its entirety to the target volume. While a volume copy has a status of In Progress, Pending, or Failed, the source volume will be available only for read activity.

After the volume copy completes, the source volume becomes available to host applications for write requests. The target volume automatically becomes read-only to hosts, and write requests to the target volume will be rejected.

The following are valid source volumes:

- Standard volume
- Snapshot volume
- Base volume of a snapshot volume
- Primary volume participating in a Remote Volume Mirror

The following are not valid source volumes:

- Secondary volume participating in a Remote Volume Mirror
- Snapshot repository volume
- Mirror repository volume
- Failed volume
- Missing volume

-
- A volume currently in a modification operation
 - A volume holding a SCSI-2 or persistent reservation
 - A volume that is a source volume or target volume in another volume copy that has a status of in progress, pending, or failed

Target Volume

CAUTION Potential data corruption. A volume copy will overwrite data on the target volume. Ensure that you no longer need the data, or have backed up the data on the target volume, before starting a volume copy.

A target volume contains a copy of the data from the source volume. The target volume can be a standard volume, the base volume of a Failed or Disabled snapshot volume, or a Remote Volume Mirror primary volume.

When a volume copy is started, data from the source volume is copied in its entirety to the target volume.

While the volume copy has a status of In Progress, Pending, or Failed, read and write requests to the target volume will be rejected by the controllers. After the volume copy operation is finished, the target volume automatically becomes read-only to hosts, and write requests to the target volume will be rejected. You can change the read-only attribute after the volume copy is completed or has been stopped. (For more information about the read-only attribute, refer to page 6-11.)

The following volumes are valid target volumes:

- Standard volume
- Base volume of a disabled or failed snapshot volume
- Primary volume participating in a Remote Volume Mirror

The following volumes are not valid target volumes:

- Base volume of an active snapshot volume
- Snapshot volume
- Mirror repository volume
- Snapshot repository volume
- Secondary volume in a Remote Volume Mirror
- Failed volume
- Missing volume

-
- A volume with a status of degraded
 - A volume currently in a modification operation
 - A volume holding a SCSI-2 or persistent reservation
 - A volume that is a source volume or target volume in another volume copy that has a status of In Progress, Pending, or Failed

Volume Copy and Persistent Reservations

You cannot use volumes holding persistent reservations for either a source volume or a target volume. Persistent reservations are configured and managed through the server cluster software and prevent other hosts from accessing the reserved volume. Unlike other types of reservations, a persistent reservation reserves host access to the volume across multiple host ports, providing various levels of access control.

To determine which volumes have reservations, run the `show (volume) reservations` command. To remove a reservation, run the `clear (volume) reservations` command.

Storage Array Performance

During a volume copy, the resources of the storage array may be diverted from processing I/O activity to completing a volume copy, which may affect the overall performance of the storage array.

Several factors contribute to the performance of the storage array, including I/O activity, volume RAID level, volume configuration (number of drives in the volume groups and cache parameters), and volume type (snapshot volumes may take more time to copy than standard volumes).

When you create a new volume copy, you will define the copy priority to determine how much controller processing time is allocated for a volume copy versus I/O activity.

Copy priority has five relative settings ranging from highest to lowest. The highest priority rate supports the volume copy, but I/O activity may be affected. The lowest priority rate supports I/O activity, but the volume copy will take longer. You define the copy priority when you create the volume copy pair. You can redefine the copy priority later using the `set volumeCopy` command. You can also redefine the volume copy priority when you recopy a volume.

Restrictions

The following restrictions apply to the source volume, target volume, and storage array:

- While a volume copy has a status of In Progress, Pending, or Failed, the source volume will be available for read activity only. After the volume copy finishes, read and write activity to the source volume will be permitted.
- A volume can be selected as a target volume for only one volume copy at a time.
- The maximum allowable number of volume copies per storage array is dependent upon the storage array configuration.
- A volume that is reserved by the host cannot be selected as a source volume or target volume.
- A volume with a status of Failed cannot be used as a source volume or target volume.
- A volume with a status of Degraded cannot be used as a target volume.
- A volume participating in a modification operation cannot be selected as a source volume or target volume. Modification operations include Dynamic Capacity Expansion (DCC), Dynamic RAID Level Migration (DRM), Dynamic Segment Sizing (DSS), Dynamic Volume Expansion (DVE), and defragmenting a volume group.

Volume Copy Commands

The Volume Copy commands and brief descriptions of what the commands do are listed in the following table:

Table 6-1 Volume Copy Commands

Command	Description
<code>create volumeCopy</code>	This command creates a volume copy and starts the volume copy operation.
<code>disable storageArray feature=volumeCopy</code>	This command turns off the current volume copy operation.
<code>enable storageArray feature</code>	This command activates the Volume Copy premium feature.
<code>recopy volumeCopy</code>	This command reinitiates a volume copy operation using an existing volume copy pair.
<code>remove volumeCopy</code>	This command removes a volume copy pair.

Command	Description
<code>set volumeCopy</code>	This command defines the properties for a volume copy pair.
<code>show volumeCopy</code>	This command returns information about volume copy operations. You can retrieve information about a specific volume copy pair, or all volume copy pairs in the storage array.
<code>show volumeCopy sourceCandidates</code>	This command returns information about the candidate volumes that you can use as the source for a volume copy operation.
<code>show volumeCopy targetCandidates</code>	This command returns information about the candidate volumes that you can use as the target for a volume copy operation.
<code>stop volumeCopy</code>	This command stops a volume copy operation.

Creating a Volume Copy

Before creating a volume copy, ensure that a suitable target volume exists on the storage array, or create a new target volume specifically for the volume copy. The target volume you can use must have a capacity equal to or greater than the source volume.

You can have a maximum of eight volume copies In Progress at one time. Any volume copy greater than eight will have a status of Pending until one of the volume copies with a status of In Progress completes.

The general steps for creating a volume copy are

1. Enable the Volume Copy premium feature.
2. Determine candidates for a volume copy.
3. Create the volume copy target and source volumes.

Enabling the Volume Copy Feature

The first step in creating a volume copy is to make sure the feature is enabled on the storage array. Because volume copy is a premium feature, you will need a feature key to enable the feature. The command for enabling the feature key file is:

```
enable storageArray feature file="filename"
```

where the `filename` parameter is the complete file path and filename of a valid feature key file. Enclose the file path and filename in double quotes (“ ”). Valid filenames for feature key files usually end with a `.key` extension.

Determining Volume Copy Candidates

All volumes and physical disks may not be available for use in volume copy operations. To determine which candidate volumes on the storage array you can use as a source volume, use the `show volumeCopy sourceCandidates` command. To determine which candidate volumes on the storage array you can use as a target volume, use the `show volumeCopy targetCandidates` command. These commands return a list of the drive module, slot, and capacity information for source volume and target volume candidates. You can use the `show volumeCopy sourceCandidates` and the `show volumeCopy targetCandidates` command only after you have enabled the volume copy feature.

Creating a Volume Copy

CAUTION Potential data corruption. A volume copy will overwrite data on the target volume. Ensure that you no longer need the data or have backed up the data on the target volume before starting a volume copy.

When you create a volume copy you must define which volumes you want to use for the source and target volumes. You define the source and target volumes by the name of each volume. You can also define the copy priority and choose whether you want the target volume to be read only after the data is copied from the source volume.

The general form of the command is:

```
create volumeCopy source="sourceName" target="targetName"  
[copyPriority=(highest | high | medium | low | lowest)  
targetReadOnlyEnabled=(TRUE | FALSE)]
```

NOTE You can use one or both optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

Before running the `create volumeCopy` command perform the following actions:

- Stop all I/O activity to the source volume and target volume.
- Unmount any file systems on the source volume and target volume, if applicable.

An example of the `create volumeCopy` command might look similar to the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "create volumecopy  
source=\"Jaba_Hut\" target=\"Obi_1\" copypriority=medium  
targetreadonlyenabled=true;"
```

The command in this example copies the data from the source volume named “Jaba_Hut” to the target volume named “Obi_1.” Setting the copy priority to “medium” provides a compromise between how quickly the data is copied from the source volume to the target volume and the amount of processing resource required for data transfers to other volumes in the storage array. Setting the `targetReadOnlyEnabled` parameter to true means that write requests cannot be made to the target volume, ensuring that the data on the target volume remains unaltered.

If you were to write this command in a script file, the command would look similar to the following example:

```
create volumecopy source="Jaba_Hut" target="Obi_1"  
copypriority=medium targetreadonlyenabled=true;
```

After the volume copy operation is completed, the target volume automatically becomes read-only to hosts. Any write requests to the target volume will be rejected, unless you

.....

disable the Read-Only attribute. You can use the `set volumeCopy` command to disable the read-only attribute.

To view the progress of a volume copy, use the `show volume actionProgress` command. This command returns information about the volume action, percentage completed, and time remaining until the volume copy is complete.

Viewing Volume Copy Properties

Using the `show volumeCopy` command, you can view information about one or more selected source volumes or target volumes. This command returns:

- Role
- Copy status
- Start timestamp
- Completion timestamp
- Copy priority
- Target volume read-only attribute setting
- Source or target volume WWN

If a volume is participating in more than one volume copy (can be a source volume for one volume copy and a target volume for another volume copy), the details are repeated for each associated copy pair.

The general form of the command is:

```
show volumeCopy (allVolumes | source [sourceName] | target [targetName])
```

An example of the `show volumeCopy` might look similar to the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show volumecopy source [\"Jaba_Hut\"];"
```

The command in this example is requesting information about the source volume “Jaba_Hut.” If you wanted information about all volumes, you would use the `allVolumes` parameter. You can also request information about a specific target volume.

If you were to write this command in a script file, the command would look similar to the following example:

```
show volumecopy source [\"Jaba_Hut\"];
```

Changing Volume Copy Settings

The `set volumeCopy` command enables you to change the property settings for a volume copy pair. Using this command you can change:

- Copy priority
- Target volume read/write permission

Copy priority has five relative settings ranging from highest to lowest. The highest priority supports the volume copy, but I/O activity may be affected. The lowest priority supports I/O activity, but the volume copy takes longer. You can change the copy priority before the volume copy begins, while the volume copy has a status of In Progress, or after the volume copy has completed when recreating a volume copy using the `recopy volumeCopy` command.

When you create a volume copy pair and after the original volume copy has completed, the target volume is automatically defined as read-only to the hosts. The read-only status of the target volume helps ensure the copied data on the target volume is not corrupted by additional writes to the target volume after the volume copy is created. You will want to maintain the read-only status when:

- You are using the target volume for backup purposes
- You are copying data from one volume group to a larger volume group for greater accessibility
- You are planning to use the data on the target volume to copy back to the base volume in case of a disabled or failed snapshot volume

At other times you may want to write additional data to the target volume. You can use the `set volumeCopy` command to reset the read/write permission for the target volume.

NOTE If you have set the volume copy parameters to enable host writes to the target volume, read and write request to the target volume will be rejected while the volume copy has a status of In Progress, Pending, or Failed.

The general form of the command is:

```
set volumeCopy target [targetName] [source [sourceName]]
copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)
```

NOTE You can use one or both parameters as needed to help define your configuration. You do not, however, need to use either parameter.

.....

An example of using the `set volumeCopy` command to change these parameters is:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "set volumecopy
target [\"Obi_1\"] copypriority=highest
targetreadonlyenabled=false;"
```

If you were to write this command in a script file, the command would look similar to the following example:

```
set volumecopy target [\"Obi_1\"] copypriority=highest
targetreadonlyenabled=false;
```

Re-Copying a Volume

Using the `recopy volumeCopy` command you can create a new volume copy for a previously defined copy pair that has a status of stopped, failed, or completed. You can use the `recopy volumeCopy` command to create backups of the target volume. You can then copy the backup to tape for off-site storage. When you use the `recopy volumeCopy` command to make a backup, you cannot write to source while the re-copy is running. The re-copy may take a long time.

When you run the `recopy volumeCopy` command the data on the source volume is copied in its entirety to the target volume.

CAUTION Potential data corruption. The `recopy volumeCopy` command overwrites existing data on the target volume and makes the target volume read-only to hosts. The `recopy volumeCopy` command will fail all snapshot volumes associated with the target volume, if any exist.

You can also reset the copy priority using the `recopy volumeCopy` command if you want to change the copy priority for the re-copy operation. The higher priorities allocate storage array resources to the volume copy at the expense of storage array performance.

The general form of the command is:

```
recopy volumeCopy target [targetName] [source [sourceName]  
copyPriority=(highest | high | medium | low | lowest)  
targetReadOnlyEnabled=(TRUE | FALSE) ]
```

NOTE You can use one or all optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

An example of the `show volumeCopy` might look similar to the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "recopy volumeCopy  
target [\"Obi_1\"] copypriority=highest;"
```

The command in this example copies data from the source volume associated with the target volume “Obi_1” to the target volume again. The copy priority is set to the highest value to complete the volume copy as quickly as possible. The underlying consideration for using this command is that you have already created the volume copy pair, which has already created one volume copy. Using this command, you are copying the data from the source volume to the target volume with the assumption that the data on the source volume was changed since the previous copy was made.

If you were to write this command in a script file, the command would look similar to the following example:

```
recopy volumeCopy target [\"Obi_1\"] copypriority=highest;
```

Stopping a Volume Copy

The `stop volumeCopy` command enables you to stop a volume copy that has a status of In Progress, Pending, or Failed. After you have stopped a volume copy, you can use the `recopy volumeCopy` command to create a new volume copy using the original volume copy pair. After you stop a volume copy, all mapped hosts will have write access to the source volume.

The general form of the command is:

```
stop volumeCopy target [targetName] [source [sourceName]]
```

An example of the `show volumeCopy` might look similar to the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "stop volumeCopy  
target [\"Obi_1\"];"
```

If you were to write this command in a script file, the command would look similar to the following example:

```
stop volumeCopy target ["Obi_1"];
```

Removing Copy Pairs

The `remove volumeCopy` command enables you to remove a volume copy pair from the storage array configuration. All volume copy-related information for the source volume and target volume is removed from the storage array configuration. The data on the source volume or target volume is not deleted. Removing a volume copy from the storage array configuration also removes the read-only attribute for the target volume.

IMPORTANT If the volume copy has a status of In Progress, you must stop the volume copy before you can remove the volume copy pair from the storage array configuration.

The general form of the command is:

```
remove volumeCopy target [targetName] [source [sourceName]]
```

An example of the `show volumeCopy` might look similar to the following example:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "remove volumeCopy  
target [\"Obi_1\"];"
```

If you were to write this command in a script file, the command would look similar to the following example:

```
remove volumeCopy target ["Obi_1"];
```

Interaction with other Features

You can run the Volume Copy premium feature while running the following premium feature:

- Storage Partitioning
- Snapshot Volumes
- Remote Volume Mirror

When you are running the Volume Copy feature with other premium features, you will need to take the requirements of other premium features into consideration to help ensure you set up a stable storage array configuration.

In addition to the premium features, you can also run the Volume Copy feature while running Dynamic Volume Expansion.

Storage Partitioning

Storage partitioning is a premium feature that enables hosts to share access to volumes in a storage array. You create a storage partition when you define:

- A host
- A host group
- A volume-to-LUN mapping

The volume to LUN mapping enables you to define which host group or host has access to a particular volume in the storage array.

After you create a volume copy, the target volume automatically becomes read-only to hosts to ensure that the data is preserved. Hosts that have been mapped to a target volume will not have write access to the volume, and any attempt to write to the read-only target volume will result in a host I/O error.

If you want hosts to have write access to the data on the target volume, use the `set volumeCopy` command to disable the Read-Only attribute for the target volume.

Snapshot Volumes

A snapshot volume is a point-in-time image of a volume and is typically created so that an application, such as a backup application, can access the snapshot volume and read the data while the base volume remains online and accessible to hosts.

.....

The volume for which the point-in-time image is created is known as the base volume and must be a standard volume in the storage array. The snapshot repository volume stores information about all data altered since the snapshot was created.

Snapshot volumes can be selected as the source volume for a volume copy. This is a good use of this feature, as it enables complete backups without significant impact to the storage array I/O. Some I/O processing resource is, however, lost to the copy operation.

IMPORTANT If you choose the base volume of a snapshot volume as your target volume, you must disable all snapshot volumes associated with the base volume before you can select it as a target volume.

When you create a snapshot volume, a snapshot repository volume is automatically created. The snapshot repository volume stores information about the data that has changed since the snapshot volume was created. Snapshot repository volumes cannot be selected as a source volume or target volume in a volume copy.

The Snapshot Volume premium feature can be used in conjunction with the Volume Copy premium feature to back up data on the same storage array, and to restore the data on the snapshot volume back to its original base volume.

Remote Volume Mirror

The Remote Volume Mirroring premium feature provides for online, real-time replication of data between storage arrays over a remote distance. In the event of a disaster or a catastrophic failure of one storage array, you can promote a secondary storage array to take over responsibility for data storage.

When you create a Remote Volume Mirror, a mirrored volume pair is created, consisting of a primary volume at the primary storage array and a secondary volume at a remote storage array.

The primary volume is the volume that accepts host I/O and stores data. When the mirror relationship is initially created, data from the primary volume is copied in its entirety to the secondary volume. This process is known as a full synchronization and is directed by the controller owner of the primary volume. During a full synchronization, the primary volume remains fully accessible for all normal I/O activity.

The controller owner of the primary volume initiates remote writes to the secondary volume to keep the data on the two volumes synchronized. Whenever the data on the primary volume and the secondary volume becomes unsynchronized, the controller owner of the primary volume initiates a re-synchronization, where only the data changed during the interruption is copied.

The secondary volume maintains a mirror of the data on its associated primary volume. The controller owner of the secondary volume receives remote writes from the primary volume controller owner, but will not accept host write requests.

.....

The secondary volume remains available to host applications as read-only while mirroring is underway. In the event of a disaster or catastrophic failure at the primary site, you can perform a role reversal to promote the secondary volume to a primary role. Hosts will then be able to access the newly promoted volume and business operations can continue.

You can select a primary volume participating in a Remote Volume Mirror to be used as the source volume or a target volume for a volume copy. A secondary volume participating in a Remote Volume Mirror cannot be selected as a source volume or target volume.

Role Reversals

A role reversal is the act of promoting the secondary volume to be the primary volume of the mirrored volume pair, and demoting the primary volume to be the secondary volume.

In the event of a disaster at the storage array containing the primary volume, you can fail over to the secondary site by performing a role reversal to promote the secondary volume to the primary volume role. This will enable hosts to continue accessing data and business operations to continue.

Attempting a role reversal in which the original primary volume is the source volume for an active volume copy (status is In Progress or Pending) causes the volume copy to fail. The failure occurs when the original primary volume becomes the new secondary volume. You cannot restart the volume copy until you return the roles of the volumes back to their original state; that is, the volume that was originally the primary volume is set once again to be the primary volume.

If the primary storage array is recovered but is unreachable due to a link failure, a forced promotion of the secondary volume will result in both the primary and secondary volumes viewing themselves in the primary volume role (dual-primary condition). If this occurs, the volume copy in which the primary volume is participating will be unaffected by the role change.

You can perform a role reversal using the `set remoteMirror` command. (For information about the `set remoteMirror` command refer to page 8-74.)

- To change a secondary volume to a primary volume:

```
set remoteMirror role=primary
```

This command promotes the selected secondary volume to become the primary volume of the mirrored pair. Use this command after a catastrophic failure has occurred.

- To change a primary volume to a secondary volume:

```
set remoteMirror role=secondary
```

This command demotes the selected primary volume to become the secondary volume. Use this command after a catastrophic failure has occurred.

Chapter 7

Maintaining a Storage Array

This chapter describes how to use the CLI and script commands to maintain a storage array.

Maintenance covers a broad spectrum of activity with the goal of keeping a storage array operational and available to all hosts. This chapter provides descriptions of commands you can use to perform storage array maintenance. The commands are organized into four sections:

- Routine Maintenance
- Performance Tuning
- Troubleshooting and Diagnostics
- Recovery Operations

The organization is not a rigid approach, and you can use the commands as appropriate for your storage array. The commands listed in this chapter do not cover the entire array of commands you can use for maintenance. Other commands, particularly the `set` commands, can provide diagnostic or maintenance capabilities.

Routine Maintenance

Routine maintenance involves those tasks you might perform periodically to ensure that the storage array is running as well as possible or to detect conditions before they become problems.

Running a Media Scan

Media scan provides a way of detecting drive media errors before they are found during a normal read or write to the physical disks. Any media scan errors detected are reported to the event log. It provides an early indication of an impending drive failure and reduces the possibility of encountering a media error during host operations. A media scan is performed as a background operation and scans all data and redundancy information in defined user volumes. A media scan runs on all volumes in the storage array that have the following conditions:

- Optimal
- Have no modification operations in progress
- Have media scan enabled

Errors detected during a scan of a user volume are reported to the major event log and handled as:

- *Unrecovered Media Error* – the physical disk could not read the requested data on its first attempt or on any subsequent retries. Result: For volumes with redundancy protection, the data is reconstructed, rewritten to the drive, verified, and the error is reported to the event log. For volumes without redundancy protection the error is not corrected, but it is reported to the event log.
- *Recovered Media Error* – the physical disk could not read the requested data on its first attempt. Result: The data is rewritten to the physical disk and verified. The error is reported to the event log.
- *Redundancy Mismatches* – redundancy errors are found and a media error is forced on the block stripe so that it is found when the physical disk is scanned again. If redundancy is repaired, this forced media error is removed. Result: The first 10 redundancy mismatches found on a volume are reported to the event log.
- *Unfixable Error* – The data could not be read and parity or redundancy information could not be used to regenerate it. For example, redundancy information cannot be used to reconstruct data on a degraded volume. Result: The error is reported to the event log.

.....

The script command `set` provides two commands to define media scan properties:

- `set volumes`
- `set storageArray`

The `set volume` command enables a media scan for the volume. The general form of the command is:

```
set (allVolumes | volume [volumeName] | volumes [volumeName1  
... volumeNameN] | volume <wwn>) mediaScanEnabled=(TRUE |  
FALSE)
```

The `set storageArray` command defines how frequently a media scan is run on a storage array. The general form of the command is:

```
set storageArray mediaScanRate=(disabled | 1-30)
```

Running a Redundancy Check

Redundancy checks are performed when media scans are run. (Refer to “Running a Media Scan” for an explanation about setting up and running media scans.) During a redundancy check, all data blocks in a volume are scanned and, depending on the RAID level, deteriorated data are corrected. Correction is performed as follows:

- RAID 3 or 5 volumes, redundancy is checked and repaired.
- RAID 1 volumes, the data is compared between the mirrored drives and data inconsistencies are repaired.
- RAID 0 volumes have no redundancy.

Before you can run a redundancy check, you must enable redundancy checking. To enable redundancy checking, use the `set volume` command. The general form of the command is:

```
set (allVolumes | volume [volumeName] | volumes [volumeName1  
... volumeNameN] | volume <wwn>)  
redundancyCheckEnabled=(TRUE | FALSE)
```

Resetting a Controller

IMPORTANT When you reset a controller, the controller is not available for I/O operations until the reset is complete. If a host is using volumes owned by the controller being reset, the I/O directed to the controller will be rejected. Before resetting the controller, either verify that the volumes owned by the controller are not in use or ensure that a multi-path driver is installed on all hosts using these volumes.

Resetting a controller is the same as rebooting the controller processors. To reset a controller, run the following command:

```
reset controller [(a | b)]
```

Enabling Controller Data Transfer

At times a controller may become quiescent while running diagnostics. If this occurs, the controller may become unresponsive. To revive a controller that has become quiescent while running diagnostics, run the following command:

```
enable controller [(a | b)] dataTransfer
```

Resetting Battery Age

After you have replaced the batteries in the storage array, you will need to reset the age of the battery. You can reset the batteries for an entire storage array or the battery in a specific controller. To reset the age of the batteries to zero days, run the following command:

```
reset storageArray batteryInstallDate [controller=(a | b)]
```

Removing Persistent Reservations

Persistent reservations preserve volume registrations and prevent hosts, other than the host defined for the volume, from accessing the volume. You must remove persistent reservations before you perform the following changes to your configuration:

- Change or delete LUN mappings on a volume holding a reservation.
- Delete volume groups or volumes that have any reservations.

To determine which volumes have reservations, run the following command:

```
show (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) reservations
```

.....

To clear persistent volume reservations, run the following command:

```
clear (allVolumes | volume[volumeName] | volumes  
[volumeName1" ... "volumeName]) reservations
```

Synchronizing Controller Clocks

To synchronize the clocks on both controllers in a storage array with the host clock, run the following command:

```
set storageArray time
```

Locating Physical Disks

At times you may need to locate a specific physical disk. In very large storage array configurations, this can sometimes be awkward. If you need to locate a specific physical disk, you can do so by turning on the LED on the front of the physical disk. To locate a physical disk, run the following command:

```
start drive [trayID,slotID] locate
```

To turn off the LED after locating the physical disk, run the following command:

```
stop drive locate
```

Performance Tuning

Over time, as a storage array exchanges data between the hosts and physical disks, its performance can degrade. You can monitor the performance of a storage array and make adjustments to the storage array operational settings to help improve performance.

Monitoring Performance

You can monitor the performance of a storage array by using the `save storageArray performanceStats`. This command saves performance information to a file that you can review to help determine how well the storage array is running. Table 7-1 lists the performance information saved to the file.

Table 7-1 Storage Array Performance Information

Type of Information	Description
Devices	Devices are: Controllers – the controller in slot A or B and a list of the volumes owned by the controller Volume – a list of the volume names Storage Array Totals – a list of the totals for both controllers in an active-active controller pair, regardless if one, both, or neither are selected for monitoring
Total I/Os	The number of total I/Os performed since the storage array was started
Read Percentage	The percentage of total I/Os that are read operations (calculate the write percentage by subtracting the read percentage from 100 percent)
Cache Hit Percentage	The percentage of reads that are fulfilled by data from the cache rather than requiring an actual read from a physical disk
Current KB/second	The current transfer rate in kilobytes per second (current means the number of kilobytes per second since the last time the polling interval elapsed, causing an update to occur)
Maximum KB/second	The highest data transfer value achieved in the current kilobyte per second statistic block
Current IO/second	The current number of I/Os per second (current means the number of I/Os per second since the last time the polling interval elapsed, causing an update to occur)

Type of Information	Description
Maximum IO/second	The highest number of I/Os achieved in the current I/O per second statistic block

The general form of the command is:

```
save storageArray performanceStats file="filename"
```

where `filename` is the name of the file in which you want to save the performance statistics. You can use any filename your operating system can support. The default file type is “.csv.” The performance information is saved as a comma delimited file.

Before you use the `save storageArray performanceStats` command, run the `set session performanceMonitorInterval` and `set session performanceMonitorIterations` commands to specify how often statistics are collected.

Changing RAID Levels

When you create a volume group, you can define the RAID level for the volumes in that group. You can later change the RAID level to improve performance or provide more secure protection for your data. To change the RAID level, run the following command:

```
set volumeGroup [volumeGroupName] raidLevel=(0|1|3|5)
```

where `volumeGroupName` is the number of the volume group for which you want to change the RAID level.

Changing Segment Size

When you create a new volume, you can define the segment size for that volume. You can later change the segment size to optimize performance. In a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy an I/O request. Using a single physical disk for a single request leaves other disks available to simultaneously service other requests. If the volume is in a single-user large I/O environment, performance is maximized when a single I/O request is serviced with a single data stripe. To change the segment size, run the following command:

```
set volume ([volumeName] | <wwn>)
segmentSize=segmentSizeValue
```

where `segmentSizeValue` is the new segment size you want. You can identify the volume by name or WWN. (Refer to the `set volume` command on page 8-82 for usage information.)

Changing Cache Parameters

The script command set provides two commands you can use to change cache parameter settings:

- `set storageArray`
- `set volume`

The `set storageArray` command enables you to change settings for

- Cache block size
- Cache flush start percentage
- Cache flush stop percentage

The `set volume` command enables you to change settings for

- Cache flush modifier
- Cache without batteries enabled/disabled
- Mirror cache enabled/disabled
- Read cache enabled/disabled
- Write cache enabled/disabled
- Read ahead multiplier

Defragmenting a Volume Group

NOTE Defragmenting a volume group starts a long running operation.

When you defragment a volume group, you are consolidating the free capacity in the volume group into one contiguous area. Defragmentation does not change the way in which the data is stored on the volumes. As an example, consider a volume group with five volumes. If you delete volumes one and three your volume group is configured as

space, volume 2, space, volume 4, volume 5, original unused space

When you defragment this group the space (free capacity) is consolidated into one contiguous location after the volumes. After being defragmented the volume group is

volume 2, volume 4, volume 5, consolidated unused space

To defragment a volume group run the following command:

```
start volumeGroup [volumeGroupName] defragment
```

where `volumeGroupName` is the identifier for the volume group.

Troubleshooting and Diagnostics

If a storage array exhibits abnormal operation or failures, you can use the commands described in this section to help determine the cause of the problems.

Collection All Support Data

To gather the most comprehensive information about a storage array, run the `save storageArray supportData` command. This command collects data for remote troubleshooting and analysis of storage management software problems. All of the files gathered are compressed into a single archive in a zipped file format. Table 7-2 lists the type of support data collected.

Table 7-2 Storage Array Support Data

Type of Data	Description and Filename
Storage Array Profile	Storage array profile data is a description of all components and properties of a storage array. storageArrayProfile.txt
Major Event Log	Major event log data is a detailed list of errors that occur on the storage array. The list is stored in reserved areas on the disks in the storage array. The list records configuration events and storage array component failures. majorEventLog.txt
Read Link Status Data	Read link status is a detailed list of errors that have been detected in the traffic flow between devices on the Fibre Channel loop. A file of historical read link stat data may also be included in the archive. readLinkStatus.csv
NVSRAM Data	NVSRAM data is a controller file that specifies the default settings for the controllers. NVSRAMdata.txt

Type of Data	Description and Filename
Performance Statistics	<p>A detailed description of how a storage array is performing. Data collected includes the I/O activity of specific controllers or volumes, the transfer rate of the controller, the current I/O per second, and the maximum I/O per second.</p> <p>performanceStatistics.csv</p>
Persistent Reservations and Registrations	<p>A detailed list of volumes on the storage array and persistent reservations and registrations</p> <p>persistentRegistrations.txt</p>
Object Bundle	<p>A detailed description of the status of your storage array and its components, valid at the time the file was generated. The object bundle file is a binary file and does not contain human readable information.</p> <p>objectBundle</p>
Drive Diagnostic Data	<p>A detailed list of log sense data from all the drives in your storage array</p> <p>driveDiagnosticData.txt</p>
Recovery Profile	<p>A detailed description of the latest recovery profile record, and historical data</p> <p>recoveryProfile.csv</p>
Unreadable Sectors	<p>A detailed list of all the unreadable sectors that have been logged to the storage arrays</p> <p>badBlocksData.txt</p>
State Capture Data	<p>A detailed description of the current state of the storage array</p> <p>stateCaptureData.dmp</p>

Collecting Physical Disk Data

To gather information about all the physical disks in a storage array, run the `save allDrives` command. This command collects sense data from all the physical disks in a storage array and saves the data to a file. The sense data consists of statistical information

.....

maintained by each of the physical disks in the storage array.

Diagnosing a Controller

The `diagnose controller` command provides three tests that enable you to verify a controller is functioning properly. The tests are:

- Read test
- Write test
- Data loopback test

The read test initiates a read command as it would be sent over an I/O data path. The read test compares data with a known, specific data pattern, checking for data integrity and errors. If the read command is unsuccessful or the data compared is not correct, the controller is considered to be in error and is placed offline.

The write test initiates a write command as it would be sent over an I/O data path to the diagnostics region on a specified drive. This diagnostics region is then read and compared to a specific data pattern. If the write fails or the data compared is not correct, the controller is considered to be in error, and is failed and placed offline.

Run the data loopback test only on controllers that have connections between the controller and the drives. The test passes data through each controller drive-side channel, minihub, out onto the loop and back again. Enough data is transferred to determine error conditions on the channel. If the test fails on any channel, this status is saved so that it can be returned if all other tests pass.

For best results, run all three tests at initial installation and any time you have made changes to the storage array or components that are connected to the storage array (such as hubs, switches, and host adapters).

A custom data pattern file called `diagnosticsDataPattern.dpf` is included on the root directory of the installation CD. You can modify this file, but the file must have the following properties to work correctly for the tests:

- The file values must be entered in hexadecimal format (00 to FF) with only one space between the values.
- The file must be no larger than 64 bytes in size. Smaller files will work, but larger files will cause an error.

The test results contain a generic, overall status message, and a set of specific test results. Each test result contains the following information:

- Test (Read/Write/Data Loopback)
- Port (Read/Write)
- Level (Internal/External)

-
- Status (Pass/Fail)

Events are written to the Event Log when Diagnostics are started and when testing is completed. These events will help you to evaluate whether diagnostics testing was successful or failed and the reason for the failure.

Running Read Link Status Diagnostics

Read link status (RLS) error counts refer to link errors that have been detected in the traffic flow of a Fibre Channel loop. The errors detected are represented as a count (32 bit field) of error occurrences accumulated over time and help to provide coarse measure of the integrity of the components and devices on the loop. By analyzing the error counts retrieved, it is possible to determine the components or devices within the Fibre Channel loop that may be experiencing problems communicating with the other devices on the loop. A high error count for a particular component or device indicates that it may be experiencing problems and should be given immediate attention.

Error counts are calculated from the current baseline. The baseline describes the error count values for each type of device in the Fibre Channel loop, either when the controller goes through its start-of-day or when you reset the baseline. The baseline indicates the difference in error counts from the time the baseline was established to the time you request the read link status data.

The script command set provides two commands for running read link status diagnostics:

- `reset storageArray RLSBaseline`
- `save storageArray RLSCounts`

The `reset storageArray RLSBaseline` command resets the read link status baseline for all devices.

The `save storageArray RLSBaseline` command saves the read link status counters to a file that you can later review.

Run the `reset storageArray RLSBaseline` command before you run the `save storageArray RLSBaseline` command.

Table 7-3 lists the type of data the file generated by the `save storageArray RLSBaseline` command contains.

Table 7-3 Storage Array RLS Baseline Data

Type of Data	Description
Devices	<p>Devices is a list of all devices on the Fibre Channel loop.</p> <p>The devices are displayed in channel order. Within each channel the devices are sorted according to the device position in the loop.</p>
Baseline Time	<p>Baseline time is the date and time when the baseline was set.</p>
Elapsed Time	<p>Elapsed time is the time that has elapsed from when the baseline time was set to when the read link status was gathered.</p>
Invalid Transmission Word (ITW)	<p>ITW data is the total number of ITW errors detected on the Fibre Channel loop from the baseline time to the current date and time. ITW may also be referred to as the “Received Bad Character Count.”</p> <p>ITW counts indicate that in decoding a read/write transmission, the mapping did not exist and the running disparity of the transmission word is invalid.</p> <p>This is the key error count to be used when analyzing the error count data.</p>
Link Failure (LF)	<p>LF data is the total number of LF errors detected on the Fibre Channel loop from the baseline time to the current date and time.</p> <p>An LF condition is either a link fault signal, a loss of signal, or a loss of synchronization condition. The LF signal indicates a failure with the media module laser operation.</p>
Loss of Synchronization (LOS)	<p>LOS data is the total number of LOS errors detected on the Fibre Channel loop from the baseline time to the current date and time.</p> <p>LOS errors indicate that the receiver cannot acquire symbol lock with the incoming data stream due to a degraded input signal. If this condition persists, the number of LOS errors will increase.</p>

Type of Data	Description
Loss of Signal (LOSG)	<p>LOSG data is the total number of LOSG errors detected on the Fibre Channel loop from the baseline date to the current date and time.</p> <p>LOSG errors typically indicate a loss of signal from the transmitting node or physical component within the Fibre Channel loop. Physical components where a loss of signal typically occurs include the GBICs, SFPs, and the Fibre Channel fibre optic cable.</p>
Primitive Sequence Protocol (PSP)	<p>PSP data is the total number of PSP errors detected on the Fibre Channel loop from the baseline date to the current date and time.</p> <p>PSP refers to the number of N_Port protocol errors detected and Link Reset Response (LRR) primitive sequences received while the link is up. A LRR is issued by another N_Port in response to a link reset.</p> <p>An N_Port is a Fibre Channel defined port at the end of a link, such as a server or workstation. Each port can act as an originator or a responder (or both) and contains a transmitter and receiver. Each port is given a unique name, called an N_Port or an NL_Port identifier. If an N_Port is connected to a loop, it becomes an NL_Port.</p> <p>An NL_Port is a Fibre Channel controller ID in a hexadecimal number. The hexadecimal number varies depending on the topology:</p> <p>For a private arbitrated loop the ID is a one-byte arbitrated loop physical address (AL_PA).</p> <p>For all other arbitrated loops, it is displayed as a single 24-bit hexadecimal number (a triplet of domain, area, and AL_PA where each field is one byte).</p> <p>For fabric and point-to-point the ID is a three-byte hexadecimal number used in the D_ID and S_ID (destination/source identifier) fields of Fibre Channel frames.</p>

Type of Data	Description
Invalid Cyclic Redundancy Check (ICRC)	<p>ICRC data is the total number of ICRC errors detected on the Fibre Channel loop from the baseline date to the current date and time.</p> <p>An ICRC count indicates that a frame has been received with an invalid cyclic redundancy check value. A cyclic redundancy check reads the data, calculates the cyclic redundancy check character, and compares the calculated cyclic redundancy check character with a cyclic check character already present in the data. If they are equal, the new data is presumed to be the same as the old data. If the calculated and old characters do not match, an error is posted and the data is resent.</p>

Interpreting the RLS Results

The way you interpret the RLS results is based on the idea that the device immediately following the problematic component will have the largest number of ITW error counts. The process is to obtain the ITW count for every component and device on the loop, analyze the data in loop order, and identify any large increases in the ITW counts.

IMPORTANT The current error counting standard for when to calculate the ITW error count is not well defined. Different vendor devices calculate at different rates. Analysis of the data must take this into consideration.

Recovery Operations

Recovery operations involve replacing failed controllers, physical disks, restoring data, and restoring the storage array to operation.

Setting Controller Operational Mode

A controller has three operational modes:

- Online
- Offline
- Service

Placing a controller online sets it to optimal and makes it active and available for I/O operations. Placing a controller offline makes it unavailable for I/O operations and moves its volume groups to the other controller if failover protection is enabled.

Taking a controller offline can seriously impact data integrity and storage array operation:

- If you do not use write cache mirroring, data in the cache of the controller you place offline will be lost.
- If you take a controller offline and you have controller failover protection through a host multi-path driver, the other controller in the pair takes over. Volume groups and their associated volumes that were assigned to the offline controller are automatically reassigned to the remaining controller. If you do not have a multi-path driver installed on the application host and you take a controller offline while the application is using associated volumes, application errors will occur.

CAUTION Potential data loss. Placing a controller offline can cause loss of data.

You would use service mode when you want to perform an operation such as replacing a controller. Placing a controller in service mode makes it unavailable for I/O operations and moves its volume groups to the second controller without affecting the volume groups preferred path. This may significantly reduce performance. The volume groups will automatically be transferred back to the preferred controller when it is placed back online.

If you change a controller to service mode while an application is using the associated volumes on the controller, the change will cause I/O errors unless a multi-path driver is installed on the host. Before you place a controller in service mode ensure:

- The volumes are not in use, or
- A multi-path driver is installed on all hosts using these volumes

In addition, if you do not have a multi-path driver, you need to make appropriate operating

.....

system specific modifications to ensure the volume groups moved when you change to service mode are accessed on the new path.

IMPORTANT Place a controller in service mode only under the direction of technical support.

To change the operational mode of a controller, run the following command:

```
set controller [(a | b)] availability=(online | offline | serviceMode)
```

Changing Controller Ownership

You can change which controller is the owner of a volume using the `set volume` command. The general form of the command is:

```
set (allVolumes | volume [volumeName] | volumes [volumeName1 ... volumeNameN] | volume <wwn>) owner=(a | b)
```

Initializing a Physical Disk

CAUTION Potential data loss. When you initialize a physical disk all data on the physical disk is lost.

You need to initialize a physical disk when you have moved physical disks that were previously part of a multidisk volume group from one storage array to another. If you do not move the entire set of physical disks, the volume group and volume information on the physical disks you move is incomplete. Each physical disk that you move contains only part of the information defined for the volume and volume group. To be able to reuse the physical disks to create a new volume group and volume, you must erase all old information from the physical disks. To erase all old information, you must initialize the physical disk.

When you initialize a physical disk, all old volume group and volume information is erased and the physical disk is returned to an unassigned state. Returning a physical disk to an unassigned state adds unconfigured capacity to a storage array. You can use this capacity to create additional volume groups and volumes.

To initialize a physical disk, run the following command:

```
start drive [trayID,slotID] initialize
```

where `trayID` and `slotID` are the identifiers for the physical disk.

Reconstructing a Physical Disk

If two or more physical disks in a volume group have failed, the volume shows a status of failed. All the volumes in the volume group are no longer operating. To return the volume group to an optimal status, you will need to replace the failed physical disks. After you have replaced the physical disks, you will need to reconstruct the data on physical disks. The data you reconstruct is the data as it would appear on the failed physical disks.

IMPORTANT You can use this command only when the physical disk is assigned to a RAID 1, 3, or 5 volume group.

To reconstruct a physical disk run the following command:

```
start drive [trayID,slotID] reconstruct
```

where `trayID` and `slotID` are the identifiers for the physical disk.

Initializing a Volume

CAUTION Potential data loss. When you initialize a volume all data on the volume and all information about the volume are destroyed.

A volume is automatically initialized when you first create it. If the volume starts exhibiting failures, you may be required to re-initialize the volume to correct the failure condition.

When you initialize a volume:

- You cannot cancel the operation once it begins.
- You cannot use this option if any modification operations are in progress on the volume or the volume group.
- You cannot change cache parameters of the volume while the initialization operation is in progress.

To initialize a volume, run the following command:

```
start volume [volumeName] initialize
```

where `volumeName` is the identifier for the volume.

Redistributing Volumes

When you redistribute volumes, you return the volumes to their preferred controller owners. The preferred controller ownership of a volume or volume group is the controller of an active-active pair that is designated to own the volumes. The preferred owner for a volume is initially designated when the volume is created. If the preferred controller is being replaced or undergoing a firmware download, ownership of the volumes is automatically shifted to the other controller, and that controller becomes the current owner of the volumes. This is considered to be a routine ownership change and is reported in the event log.

To redistribute volumes to their preferred controllers, run the following command:

```
reset storageArray volumeDistribution
```

where `volumeName` and `wwn` are the identifiers for the volumes. You can run this command on more than one volume.

IMPORTANT If you are running this command without a multi-path driver on the hosts, stop I/O activity to the volumes to prevent application errors.

IMPORTANT You cannot run this command if all volumes are currently owned by their preferred controller, or the storage array does not have defined volumes.

Under some application host operating system environments, you must reconfigure the multi-path host driver. You may also need to make operating system modifications to recognize the new I/O path to the volume.

.....

Chapter 8

Script Commands

This chapter describes the script commands that you can use to configure, monitor, and maintain a storage array. This chapter is organized in four sections:

- The first section lists general formatting rules that apply to the command syntax.
- The second section explains how to interpret the firmware level information.
- The third section lists the commands alphabetically and, for each command, includes: command name, syntax, and parameters
- The fourth section lists the commands by functional activity:
 - Storage array
 - Controller
 - Physical Disk
 - Module
 - Volume group
 - Volume
 - Host topology
 - Snapshot
 - Remote Volume Mirroring
 - Volume copy
 - Session
 - Uncategorized

CAUTION The command line interface is capable of damaging a configuration and causing loss of data if not used properly. Command operations are performed as soon as you run the commands. Some commands can immediately delete configurations or data. Before using the command line interface, make sure you have backed up all data, and save the current configuration so you can re-install it if changes you make do not work.

Command Formatting Rules

This section describes the general rules for formatting a script command and how the command syntax is presented in the following command descriptions. Syntax unique to a specific command is explained in the “Notes” section at the end of each command description.

- The script commands are not case sensitive. You can type the commands in lower case, upper case, or mixed case. (In the following command descriptions, mixed case is used as an aid to reading the command names and understanding the purpose of the command.)
- You must enter spaces in the commands as they are shown in the command descriptions.
- Square brackets are used in two ways:
 - As part of the command syntax
 - To indicate the parameters are optional

The description of each parameter will tell you if you need to enclose a parameter value in square brackets.

- Parentheses shown in the command syntax enclose specific choices for a parameter; that is, if you want to use the parameter you must enter one of the values enclosed in parentheses. Generally, you will not include parentheses in a command; however, in some instances, when you enter lists you must enclose the list in parentheses. Such a list might be a list of tray ID and slotID values. The description of each parameter will tell you if you need to enclose a parameter value in parentheses.
- Vertical bars in a command indicate “or” and separate the valid entries for the parameter. For example, the syntax for the `raidLevel` parameter in the command description appears as:

```
raidLevel=(0 | 1 | 3 | 5)
```

To use the `raidLevel` parameter to set a RAID level of “5” enter:

```
raidLevel=5
```

- When you specify physical disk locations using “trayID” and “slotID” values, separate the ID values by a comma. If you are entering more than one set of ID values, separate each set of values by a space. Enclose the set of values in parentheses. For example:

```
(1,1 1,2 1,3 1,4 2,1 2,2 2,3 2,4)
```

-
- Italic terms in the command indicate a value or information that you need to provide. For example, when you encounter the italic term:

numberOfDrives

replace the italic term with a value for the number of physical disks you want to include with the command.

- You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names of the following components:

- Storage arrays
- Host groups
- Hosts
- Volume groups
- Volumes
- Host port

Names can have a maximum of 30 characters. If the label contains more than one word, white spaces, underscores, or dashes, enclose the name in double quotes. In some usage, you must also surround the name with square brackets. The description of each parameter will indicate whether you need to enclose a parameter in double quotes or square brackets. The character string cannot contain a new line. Make sure you use unique names; if you do not use unique names the controller firmware will return an error.

When you enter a world-wide name (WWN) of a host port, some usage requires you to surround the WWN with double quotes. In other uses you must surround the WWN with angle brackets (< >). The description of the WWN parameter will indicate whether you need to enclose the WWN in double quotes or angle brackets.

- Script commands must end with a semicolon (;). You can enter more than one script command on the command line each time you enter a CLI command.

Firmware Compatibility levels

The script commands and command parameters do not run under all versions of the controller firmware. The script commands in the following section list the minimum firmware levels under which the commands can run. In the commands, the firmware levels are listed under the heading “Minimum Firmware Level.” The following list explains how to interpret the information about the firmware levels.

- If a command does not list a minimum firmware level, the command and all parameters associated with that command can run under any level of firmware.
- A firmware number without any explanatory information indicates that the firmware level applies to the entire command and all parameters for that command.
- A firmware number associated with a parameter indicates the minimum firmware level under which the parameter can run.

Examples

The `create hostGroup` command has the following section:

Minimum Firmware Level

5.20

This indicates that the entire command runs under firmware versions 5.20 or higher.

The `show volume` command has the following section:

Minimum Firmware Level

5.00

`summary` parameter 5.43

This indicates that the command and all parameters except `summary` run under firmware versions 5.00 or higher. The `summary` parameter runs under firmware versions 5.43 or higher.

Commands Listed Alphabetically

Activate Remote Volume Mirroring Feature

This command creates the mirror repository volume and activates the Remote Volume Mirror feature. When you use this command, you can define the repository volume in one of three ways: user-defined physical disks, user-defined volume group, or user-defined number of physical disks for the repository volume. If you choose to define a number of physical disks the controller firmware chooses which physical disks to use for the repository volume.

Syntax (User-Defined Physical Disks)

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5)
repositoryDrives=(trayID1,slotID1 ... trayIDn,slotIDn)
[trayLossProtect=(TRUE | FALSE)]
```

Syntax (User-Defined Volume Group)

```
activate storageArray feature=remoteMirror
repositoryVolumeGroup=volumeGroupNumber
[repositoryFreeCapacityArea=freeCapacityValue
trayLossProtect=(TRUE | FALSE)]
```

Syntax (User-Defined Number of Physical Disks)

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5)
repositoryDriveCount=numberOfDrives [driveType=(fibre | SATA
| PATA) trayLossProtect=(TRUE | FALSE)]
```

Parameters

Parameter	Description
<code>repositoryRAIDLevel</code>	The RAID level for the repository volume. Valid values are 1 , 3 , or 5 .
<code>repositoryDrives</code>	The physical disks for the repository volume. Specify the tray ID and slot ID for each physical disk you assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in parentheses.
<code>repositoryVolumeGroup</code>	The sequence number of the volume group where the repository volume will be located. (To determine the sequence numbers of the volume groups in your storage array, enter the <code>show storageArray profile</code> command.)
<code>repositoryFreeCapacityArea</code>	The free space in an existing volume group that you want to use to create the mirror repository volumes. Free capacity area is defined as the free capacity between existing volumes in a volume group. For example, a volume group might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3 free capacity. To use the free capacity following volume 2, you would specify: <code>repositoryFreeCapacityArea=2.</code>
<code>repositoryDriveCount</code>	The number of unassigned physical disks that you want to use for the repository volume.
<code>driveType</code>	The type of physical disks that you want to use for the repository volume. Valid physical disk types are: Fibre , SATA , or PATA . Use this parameter only when you use the <code>repositoryDriveCount</code> parameter. You need to use this parameter only when you have more than one type of drive in your storage array.
<code>trayLossProtect</code>	Specifies that tray loss protection will be enforced when creating the repository. To enforce tray loss protection, set this parameter to TRUE . The default setting is FALSE .

Notes

IMPORTANT If the drives you select for the `repositoryDrives` parameter are not compatible with other parameters (such as the `repositoryRAIDLevel` parameter), the command will return an error, and Remote Volume Mirroring will not be activated. The error returns the amount of space needed for the repository volume. You can then re-enter the command specifying the appropriate amount of space.

If you enter a value for the repository storage space that is too small for the mirror repository volumes, the firmware will return an error message giving the amount of space needed for the mirror repository volumes. The command will not attempt to activate the Remote Volume Mirror. You can re-enter the command using the value from the error message for the repository storage space value.

When you assign the physical disks, if you set `trayLossProtect` to true and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect` to false, the storage array will perform operations, but the volume group you create may not have tray loss protection.

When the controller firmware assigns the physical disks, if you set `trayLossProtect` to true the storage array will error the function if the controller firmware cannot provide drives that will result in the new volume group having tray loss protection. If you set `trayLossProtect` to false the storage array will perform the operation even if it means the volume group may not have tray loss protections.

Minimum Firmware Level

6.10

Activate Storage Array Firmware

This command activates firmware that you have previously downloaded to the pending configuration area on the controllers in the storage array.

Syntax

```
activate storageArray firmware
```

Parameters

None

Minimum Firmware Level

6.10

Autoconfigure Storage Array

This command automatically configures a storage array. Before entering the `autoConfigure storageArray` command, enter the `show storageArray autoConfiguration` command. The `show storageArray autoConfigure` command returns configuration information in the form of a list of valid physical disk types, RAID levels, volume information, and hotspare information. (This list corresponds to the parameters for the `autoConfigure storageArray` command.) The controllers audit the storage array and then determine the highest RAID level the storage array can support and the most efficient volume definition for the RAID level. If the configuration described by the returned list is acceptable, you can enter the `autoConfigure storageArray` command without any parameters. If you want to modify the configuration, you can change the parameters to meet your configuration requirements. You can change a single parameter or all parameters. After you enter the `autoConfigure storageArray` command, the controllers set up the storage array using either the default parameters or those you selected.

Syntax

```
autoConfigure storageArray [driveType=(fibre | SATA | PATA)
raidLevel=(0 | 1 | 3 | 5) volumeGroupWidth=numberOfDrives
volumeGroupCount=numberOfVolumeGroups
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotspares segmentSize=segmentSizeValue
readAheadMultiplier=multiplierValue]
```

Parameters

Parameter	Description
<code>driveType</code>	The type of physical disks that you want to use for the storage array. Valid physical disk types are: Fibre , SATA , or PATA . The <code>driveType</code> parameter is not required if only one type of physical disk is in the storage array.
<code>raidLevel</code>	The RAID level of the volume group that contains the physical disks in the storage array. Valid RAID levels are 0 , 1 , 3 , or 5 .

Parameter	Description
volumeGroupWidth	The number of physical disks in a volume group in the storage array.
volumeGroupCount	The number of volume groups in the storage array. Use integer values.
volumesPerGroupCount	The number of equal capacity volumes per volume group. Use integer values.
hotSpareCount	The number of hot spares you want in the storage array. Use integer values.
segmentSize	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.
readAheadMultiplier	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.

Notes

A volume group is a set of physical disks logically grouped together by the controllers in the storage array. The number of physical disks in a volume group depends on the size of physical disks available for the volume group. The following list relates physical disk capacity to the number of physical disks in a volume group:

- 18 and 36 GB physical disks, 30 physical disks maximum in the volume group
- 73 GB physical disks, 28 physical disks maximum in the volume group
- 180 GB physical disks, 11 physical disks maximum in the volume group

A 73 or 180 GB physical disk will not report exactly 73 or 180 GB as its capacity. The limitation for the number of large physical disks in a volume group is the two terabyte size restriction.

Hot spare physical disks can replace any failed physical disk in the storage array. The hot spare must be the same type of physical disk as the physical disk that failed (that is, a SATA hot spare cannot replace a Fibre Channel physical disk). A hot spare must have capacity greater than or equal to any physical disk that can fail. If a hot spare is smaller than a failed physical disk, the hot spare cannot be used to rebuild the data from the failed physical disk. Hot spares are available only for RAID level 1, 3, or 5. The maximum number of SCSI hot spares is equal to the number of SCSI physical disk channels supported by the controller. The maximum number of Fibre Channel hot spares per storage array is 15.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block

.....

stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfers.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Minimum Firmware Level

6.10

Autoconfigure Storage Array Hot Spares

This command automatically defines and configures the hot spares in a storage array. You can run this command at any time. This command provides the best hot spare coverage for a storage array.

Syntax

```
autoConfigure storageArray hotSpares
```

Parameters

None.

Notes

When you run the autoconfigure hot spares command the controller firmware determines the number of hot spares to create based on the total number and type of physical disks in the storage array. For Fibre and SATA physical disks: firmware creates 1 hot spare for the storage array and 1 additional hot spare for every 60 physical disks in the storage array. For PATA physical disks: firmware creates 1 hot spare for the storage array and 1 additional hot spare for every 20 physical disks in the storage array.

Minimum Firmware Level

6.10

Check Volume Parity

This command checks a volume for parity and media errors, and writes the results of the check to a file.

Syntax

```
check volume [volumeName] parity [parityErrorFile=filename]  
[mediaErrorFile=filename] [priority=(highest | high | medium  
| low | lowest)] [verbose=(TRUE|FALSE)]
```

Parameters

Parameter	Description
volumeName	The name of the specific volume for which you want to check parity. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
parityErrorFile	The name of the file in which you want to save the parity error information. Enclose the filename in double quotes (“ ”).
mediaErrorFile	The name of the file in which you want to save the media error information. Enclose the filename in double quotes (“ ”).
priority	Defines the priority the parity check will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest .
verbose	Captures progress details, such as percent complete, and displays the information as volume parity is being repaired. To capture progress details, set this parameter to TRUE . To prevent capturing progress details, set this parameter to FALSE .

Minimum Firmware Level

6.10

Clear Drive Channel Statistics

This command resets the statistics for all physical disk channels.

Syntax

```
clear allDriveChannels stats
```

Parameters

None.

Minimum Firmware Level

6.10

Clear Storage Array Configuration

This command clears the entire configuration from the controllers in a storage array. Information defining all volume groups, volumes, and hot spares is deleted. Use this command when you need to create a new configuration on a storage array that already has a configuration defined.

CAUTION **Potential storage array configuration damage.** As soon as you run this command, the existing storage array configuration is deleted.

Syntax

```
clear storageArray configuration
```

Parameters

None.

Minimum Firmware Level

6.10

Clear Storage Array Event Log

This command clears the storage array event log by deleting the data in the event log buffer.

CAUTION: Potential storage array configuration damage. As soon as you run this command, the existing storage array event log is deleted.

Syntax

```
clear storageArray eventLog
```

Parameters

None.

Minimum Firmware Level

6.10

Clear Storage Array Firmware Pending Area

This command deletes from the pending area buffer a firmware image or NVSRAM values you have previously downloaded.

CAUTION: Potential storage array configuration damage. As soon as you run this command, the contents of the existing storage array pending area are deleted.

Syntax

```
clear storageArray firmwarePendingArea
```

Parameters

None.

Minimum Firmware Level

6.10

Clear Volume Reservations

This command clears persistent volume reservations.

Syntax

```
clear (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) reservations
```

Parameters

Parameter	Description
allVolumes	Use this parameter to clear reservations on all volumes in the storage array.
volumeName	The name of the specific volume for which you want to clear reservations. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

5.40

Clear Volume Unreadable Sectors

This command clears unreadable sector information from one or more volumes.

Syntax

```
clear (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) unreadableSectors
```

.....

Parameters

Parameter	Description
allVolumes	Use this parameter to clear unreadable sector information from all volumes in the storage array.
volumeName	The name of the specific volume for which you want to clear unreadable sector information. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

6.10

Create Host

This command creates a new host. If you do not specify a host group in which to create the new host, it is created in the default group.

Syntax

```
create host userLabel="hostName" [hostGroup= ("hostGroupName"  
| defaultGroup)]
```

Parameters

Parameter	Description
<code>userLabel</code>	The name you want to give the host you are creating. Enclose the host name in double quotes (“ ”).
<code>hostGroup</code>	The name of the host group in which you want to create a new host. Enclose the host group name in double quotes (“ ”). (If a host group does not exist you can create a new host group using the <code>create hostGroup</code> command.) <code>defaultGroup</code> is the host group that contains the host to which the volume is mapped.

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

A host is a computer that is attached to the storage array and accesses the volumes on the storage array through its host ports (host adapters). You can define specific volume-to-LUN mappings to an individual host or assign the host to a host group that shares access to one or more volumes.

A host group is an optional topological element that you can define if you want to designate a collection of hosts that will share access to the same volumes. The host group is a logical entity. Define a host group only if you have two or more hosts that will share access to the same volumes.

If you do not specify a host group in which to place the host you are creating, the newly defined host will belong to the default host group.

Minimum Firmware Level

5.20

Create Host Group

This command creates a new host group.

Syntax

```
create hostGroup userLabel="hostGroupName"
```

.....

Parameters

Parameter	Description
hostGroupName	The name you want to give the host group you are creating. Enclose the host name in double quotes (“ ”).

Notes

A host group is an optional topological element that you can define if you want to designate a collection of hosts that will share access to the same volumes. The host group is a logical entity. Define a host group only if you have two or more hosts that will share access to the same volumes.

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

5.20

Create Host Port

This command creates a new host port.

Syntax

```
create hostPort identifier="wwn" userLabel="portLabel"  
host="hostName" [hostType=(hostTypeIndexLabel |  
hostTypeIndexNumber) ]
```

Parameters

Parameter	Description
identifier	The world wide name (<i>wwn</i>) of the host port. Enclose the wwn name in double quotes (“ ”).
portLabel	The name you want to give the new host port. Enclose the port label in double quotes (“ ”).

Parameter	Description
hostName	The name of the host for which you are defining a host port. Enclose the host name in double quotes (“ ”).
hostType	The index label or number of the host port host type. Use the <code>show storageArray hostTypeTable</code> command to generate a list of available host type identifiers. If the host type has more than one word, white spaces, or special characters, enclose the host type in double quotes (“ ”).

Notes

A host port is a physical connection on a host adapter that resides within a host computer. A host port provides a host access to the volumes in a storage array. If the host adapter has only one physical connection (one host port) the terms host port and host adapter are synonymous.

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

5.20

Create RAID Volume (Automatic Drive Select)

This command creates a volume group across the storage array physical disks, and a new volume in the volume group. The storage array controllers choose the physical disks to be included in the volume.

Syntax

```
create volume driveCount=numberOfDrives raidLevel=(0 | 1 | 3
| 5) userLabel="volumeName" [driveType=(fibre | SATA | PATA)
capacity=volumeCapacity owner=(a | b)
readAheadMultiplier=multiplierValue
segmentSize=segmentSizeValue trayLossProtect=(TRUE | FALSE)]
```

.....

Parameters

Parameter	Description
<code>driveCount</code>	The number of unassigned physical disks that you want to use in the volume group.
<code>raidLevel</code>	The RAID level of the volume group that contains the volume. Valid values are 0 , 1 , 3 , or 5 .
<code>userLabel</code>	The name you want to give the new volume. Enclose the new volume name in double quotes (“ ”).
<code>driveType</code>	Defines the type of physical disk you want to use in the volume. You cannot mix physical disk types. Valid physical disk types are: Fibre , SATA , or PATA .
<code>capacity</code>	The size of the volume you are adding to the storage array. Size is defined in units of bytes, KB, MB, or GB.
<code>owner</code>	The controller that owns the volume. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
<code>readAheadMultiplier</code>	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.
<code>segmentSize</code>	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.
<code>trayLossProtect</code>	Specifies that tray loss protection will be enforced when creating the volume group. To enforce tray loss protection, set this parameter to TRUE . The default setting is FALSE .

Notes

The `driveCount` parameter enables you to choose the number of physical disks you want to use in the volume group. You do not have to specify the physical disks by tray ID and slot ID. The controllers choose the specific physical disks to use for the volume group.

The `owner` parameter defines which controller owns the volume.

.....

If you do not specify a capacity, all physical disk capacity available in the volume group is used. If you do not specify capacity units, bytes are used as the default.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfer.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Cache read-ahead enables the controller to copy additional data blocks into cache while the controller reads and copies host requested data blocks from disk into cache. This increases the chance that a future request for data can be fulfilled from cache. Cache read-ahead is important for multimedia applications that use sequential data transfers. Choosing a higher cache read-ahead multiplier might increase the cache hit percentage; however, transfer time might also increase. The read-ahead multiplier value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller will return a list of appropriate values.

For tray loss protection to work, each physical disk in a volume group must be on separate module. If you set `trayLossProtect` to true and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect` to false, the storage array will perform operations, but the volume group you create may not have tray loss protection.

Tray loss protection is not valid when creating volumes on existing volume groups.

Minimum Firmware Level

5.20

Create RAID Volume (Free Capacity Base Select)

This command creates a volume in the free space of a volume group.

.....

Syntax

```
create volume volumeGroup=volumeGroupName  
userLabel="volumeName" [freeCapacityArea=freeCapacityValue  
capacity=volumeCapacity owner=(a | b)  
readAheadMultiplier=multiplierValue  
segmentSize=segmentSizeValue]
```

Parameters

Parameter	Description
volumeGroup	The sequence number of the volume group in which you want to create the new volume. (To determine the sequence numbers of the volume groups in your storage array, enter the <code>show storageArray Profile</code> command.)
userLabel	The name you want to give the new volume. Enclose the new volume name in double quotes (“ ”).
freeCapacityArea	The free space in an existing volume group that you want to use to create the new volume. Free capacity is defined as the free capacity between existing volumes in a volume group. For example, a volume group might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3 free capacity. To use the free capacity following volume 2, you would specify: <code>freeCapacityArea=2.</code>
capacity	The size of the volume you are adding to the storage array. Size is defined in units of bytes, KB, MB, or GB.
owner	The controller that owns the volume. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
readAheadMultiplier	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.

Parameter	Description
segmentSize	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

The `owner` parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the volume group.

If you do not specify a capacity, all available capacity in the free capacity area of the volume group is used. If you do not specify capacity units, bytes are used as the default.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfers.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Cache read-ahead enables the controller to copy additional data blocks into cache while the controller reads and copies host requested data blocks from disk into cache. This increases the chance that a future request for data can be fulfilled from cache. Cache read-ahead is important for multimedia applications that use sequential data transfers. Choosing a higher cache read-ahead multiplier might increase the cache hit percentage, however, transfer time might also increase. The read-ahead multiplier value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller will return a list of appropriate values.

Minimum Firmware Level

5.20

Create RAID Volume (Manual Drive Select)

This command creates a new volume group and volume, and enables you to specify the physical disks for the volume.

IMPORTANT You cannot use mixed physical disk types in the same volume group and volume. This command will fail if you specify different types of physical disks for the RAID volume.

Syntax

```
create volume drives=(trayID1,slotID1...trayIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5) userLabel="volumeName"
[capacity=volumeCapacity owner=(a | b)
readAheadMultiplier=multiplierValue
segmentSize=segmentSizeValue trayLossProtect=(TRUE | FALSE)]
```

Parameters

Parameter	Description
drives	Defines the physical disks you want to assign to the volume you want to create. Specify the tray ID and slot ID for each unassigned physical disk you want to assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values with parentheses.
raidLevel	The RAID level of the volume group that contains the volume. Valid values are 0 , 1 , 3 , or 5 .
userLabel	The name you want to give the new volume. Enclose the new volume name in double quotes (" ").
capacity	The size of the volume you are adding to the storage array. Size is defined in units of bytes, KB, MB, GB, or TB.
owner	The controller that owns the volume. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.

Parameter	Description
<code>readAheadMultiplier</code>	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.
<code>segmentSize</code>	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.
<code>trayLossProtect</code>	Specifies that tray loss protection will be enforced when creating the repository. To enforce tray loss protection, set this parameter to TRUE . The default setting is FALSE .

Notes

If you set the `raidLevel` parameter to RAID 1, the controller firmware takes the list of physical disks and pairs them using the following algorithm:

Data drive = X

Parity drive = $N/2+X$

where X goes from 1 to $N/2$ and N is the number of physical disks in the list. For example, if you have six physical disks the mirror pairs are

Data	Parity
1	$N/2+1=4$
2	$N/2+2=5$
3	$N/2+3=6$

You can use any combination of alphanumeric characters and the underscore (`_`) and dash (`-`) characters for the names.

The `owner` parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the volume group.

If you do not specify a capacity, all physical disk capacity available in the volume group is used. If you do not specify capacity units, bytes are used as the default.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block

.....

stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfer.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Cache read-ahead enables the controller to copy additional data blocks into cache while the controller reads and copies host requested data blocks from disk into cache. This increases the chance that a future request for data can be fulfilled from cache. Cache read-ahead is important for multimedia applications that use sequential data transfers. Choosing a higher cache read-ahead multiplier might increase the cache hit percentage, however, transfer time might also increase. The read-ahead multiplier value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller will return a list of appropriate values.

For tray loss protection to work, each physical disk in a volume group must be on separate module. If you set `trayLossProtect` to true and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect` to false, the storage array will perform operations, but the volume group you create may not have tray loss protection.

Tray loss protection is not valid when creating volumes on existing volume groups.

Minimum Firmware Level

5.20

Create Remote Volume Mirror

This command creates both the primary and secondary volumes for a Remote Volume Mirror. This command also sets the write mode (synchronous or asynchronous) and synchronization priority.

Syntax

```
create remoteMirror primary="primaryVolumeName"  
secondary="secondaryVolumeName"  
(remoteStorageArrayName="storageArrayName" |  
remoteStorageArrayWwn="wwn") [remotePassword="password"  
syncPriority=(highest | high | medium | low | lowest)  
writeOrder=(preserved | notPreserved) writeMode=(synchronous  
| asynchronous)]
```

Parameters

Parameter	Description
primary	The name of an existing volume on the local storage array you want to use for the primary volume. Enclose the primary volume name in double quotes (“ ”).
secondary	The name of an existing volume on the remote storage array you want to use for the secondary volume. Enclose the secondary volume name in double quotes (“ ”).
remoteStorageArrayName	The name of the remote storage array. Enclose the remote storage array name in double quotes (“ ”).
remoteStorageArrayWwn	The world wide name of the remote storage array. Enclose the world wide name in double quotes (“ ”).
remotePassword	The password for the remote storage array. Use when the remote storage array is password protected. Enclose the password in double quotes (“ ”).
syncPriority	Defines the priority full synchronization will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest
writeOrder	Defines write order for data transmission between the primary volume and secondary volume. Valid values are preserved or notPreserved .

Parameter	Description
writeMode	Defines how the primary volume writes to the secondary volume. Valid values are synchronous or asynchronous .

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

When choosing the primary and secondary volumes, the secondary volume must be of equal or greater size than the primary volume. The RAID level of the secondary volume does not have to be the same as the primary volume.

For a 2882 controller, you can define 32 remote mirrors. For the 4884 and 5884 controllers, you can define 64 remote mirrors.

Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of alpha-numeric characters with a maximum of 30 characters. (You can define a storage array password using the `set storageArray` command.)

Synchronization priority defines the amount of system resources used to synchronize the data between the primary and secondary volumes of a mirror relationship. If you select the highest priority level, the data synchronization uses the most system resource to perform the full synchronization, which will decrease performance for host data transfers.

The `writeOrder` parameter applies only to asynchronous mirrors and makes them become part of a consistency group. Setting the write order to **preserved** causes the mirrored pair to transmit data from the primary to secondary in the same order as the host writes to the primary. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This can require additional system overhead to maintain the buffered data, slowing operations. Setting write order to **notPreserved** frees the system from having to maintain data in a buffer, but requires forcing a full synchronization to ensure the secondary volume has the same data as the primary.

Minimum Firmware Level

6.10

Create Snapshot Volume

This command creates a snapshot volume. When you use this command, you can define the snapshot volume in one of three ways: user-defined physical disks, user-defined volume group, or user-defined number of physical disks for the snapshot volume. If you choose to define a number of physical disks, the controller firmware chooses which physical disks to use for the snapshot volume.

Syntax (User-Defined Physical Disks)

```
create snapshotVolume baseVolume="baseVolumeName"  
[repositoryRAIDLevel=(1 | 3 | 5)  
repositoryDrives=(trayID1, slotID1 ... trayIDn, slotIDn)  
userLabel="snapshotVolumeName"  
warningThresholdPercent=percentValue  
repositoryPercentOfBase=percentValue  
repositoryUserLabel="repositoryName"  
repositoryFullPolicy=(failBaseWrites | failSnapShot)  
trayLossProtect=(TRUE | FALSE)]
```

Syntax (User-Defined Volume Group)

```
create snapshotVolume baseVolume="baseVolumeName"  
[repositoryVolumeGroup=volumeGroupNumber  
repositoryFreeCapacityArea=freeCapacitySize]  
[userLabel="snapshotVolumeName"  
warningThresholdPercent=percentValue  
repositoryPercentOfBase=percentValue  
repositoryUserLabel="repositoryName"  
repositoryFullPolicy=(failBaseWrites | failSnapShot)  
trayLossProtect=(TRUE | FALSE)]
```

Syntax (User-Defined Number of Physical Disks)

```
create snapshotVolume baseVolume="baseVolumeName"  
[repositoryRAIDLevel=(1 | 3 | 5)  
repositoryDriveCount=numberOfDrives driveType=(fibre | SATA  
| PATA) userLabel="snapshotVolumeName"  
warningThresholdPercent=percentValue  
repositoryPercentOfBase=percentValue  
repositoryUserLabel="repositoryName"  
repositoryFullPolicy=(failBaseWrites | failSnapShot)  
trayLossProtect=(TRUE | FALSE)]
```

Parameters

Parameter	Description
<code>baseVolume</code>	The name of the base volume from which you want to take a snapshot. Enclose the base volume name in double quotes (“ ”).
<code>repositoryRAIDLevel</code>	The RAID level for the repository volume. Valid values are 1 , 3 , or 5 .
<code>repositoryDrives</code>	Defines the physical disks you want to assign to the repository. Specify the tray ID and slot ID for each physical disk you assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the list of repository physical disks in parentheses.
<code>repositoryDriveCount</code>	The number of unassigned physical disks that you want to use for the repository volume.
<code>driveType</code>	The type of physical disks that you want to use for the repository volume. Valid physical disk types are: Fibre , SATA , or PATA .
<code>repositoryVolumeGroup</code>	The sequence number of the volume group where the repository volume is located.
<code>repositoryFreeCapacityArea</code>	The free space in an existing volume group that you want to use to create the snapshot repository volume. Free capacity is defined as the free capacity between existing volumes in a volume group. For example, a volume group might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3 free capacity. To use the free capacity following volume 2, you would specify: <code>repositoryFreeCapacityArea=2.</code>
<code>userLabel</code>	The name you want to give the snapshot volume. Enclose the snapshot volume name in double quotes (“ ”).
<code>warningThresholdPercent</code>	The percentage of repository capacity at which you will receive a warning that the repository is nearing full. Use integer values. For example, a value of 70 means 70 percent. The default value is 50 percent.

Parameter	Description
<code>repositoryPercentOfBase</code>	The size of the repository volume as a percentage of the base volume. Use integer values. For example, a value of 40 means 40 percent. The default value is 20 percent.
<code>repositoryUserLabel</code>	The name you want to give to the repository volume. Enclose the repository volume name in double quotes (“ ”).
<code>repositoryFullPolicy</code>	Defines how you want snapshot processing to continue if the repository is full. You can choose to fail writes to the base volume (<code>failBaseWrites</code>) or fail the snapshot volume (<code>failSnapShot</code>). The default is <code>failSnapShot</code> .
<code>trayLossProtect</code>	Specifies that tray loss protection will be enforced when creating the repository. To enforce tray loss protection, set this parameter to TRUE . The default setting is FALSE .

Notes

A snapshot volume is a point-in-time image of a volume in a storage array. The base volume is the volume from which you take a “snapshot.” The volume you are taking a snapshot of must be a standard volume in the storage array. The maximum number of snapshot volumes you can create is one half of the total number of volumes supported by a controller.

You can use any combination of alphanumeric characters and the underscore (`_`) and dash (`-`) characters for the names.

One technique for naming the snapshot volume and the repository volume is to add a hyphenated suffix to the original base volume name. The suffix distinguishes between the snapshot volume and the repository volume. For example: If you have a base volume with a name of “Engineering Data” the snapshot volume can have a name of “Engineering Data-S1” and the repository volume can have a name of “Engineering Data-R1.”

If you do not choose a name for the either the snapshot volume or repository volume, the controllers create a default name using the base volume name. An example of the snapshot volume name the controllers might create is: if the base volume name is “aaa” and does not have a snapshot volume, the default snapshot volume name is “aaa-1.” If the base volume already has n-1 number of snapshot volumes, the default name is “aaa-n.” An example of the repository volume name the controller might create is: if the base volume name is “aaa” and does not have a repository volume, the default repository volume name is “aaa-R1.” If the base volume already has n-1 number of repository volumes, the default name is “aaa-Rn.”

.....

If you do not specify the unconfigured or free space, the repository volume is placed in the same volume group as the base volume. If the volume group where the base volume resides does not have enough space this command will fail.

When you assign the physical disks, if you set `trayLossProtect` to true and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect` to false, the storage array will perform operations, but the volume group you create may not have tray loss protection.

When the controller firmware assigns the physical disks, if you set `trayLossProtect` to true the storage array will error the function if the controller firmware cannot provide drives that will result in the new volume group having tray loss protection. If `trayLossProtect` to false the storage array will perform the operation even if it means the volume group may not have tray loss protections.

Minimum Firmware Level

5.00

Create Volume Copy

This command creates a volume copy and starts the volume copy operation.

IMPORTANT You can have a maximum of eight volume copies in progress at one time. If you try to create more than eight volume copies at one time, the controllers will return a status of “Pending” until one of the volume copies that is in progress finishes and returns a status of complete.

Syntax

```
create volumeCopy source="sourceName" target="targetName"
[copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)]
```

Parameters

Parameter	Description
source	The name of an existing volume you want to use as the source volume. Enclose the source volume name in double quotes (“ ”).
target	The name of an existing volume you want to use as the target volume. Enclose the target volume name in double quotes (“ ”).
copyPriority	Defines the priority volume copy will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest .
targetReadOnlyEnabled	Defines whether you can write to the target volume or only read from the target volume. To be able to write to the target volume, set this parameter to FALSE . To prevent writing to the target volume, set this parameter to TRUE .

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Copy priority defines the amount of system resources used to copy the data between the source and target volumes of a volume copy pair. If you select the highest priority level, the volume copy uses the most system resource to perform volume copy, which will decrease performance for host data transfers.

Minimum Firmware Level

5.40

Deactivate Remote Volume Mirror

Purpose

This command deactivates the Remote Volume Mirror feature and tears down the mirror repository volume. The host port dedicated to the Remote Volume Mirror is made available for host data transfers.

.....

Syntax

```
deactivate storageArray feature=remoteMirror
```

Parameters

None.

Minimum Firmware Level

6.10

Delete Host

This command deletes a host.

Syntax

```
delete host [hostName]
```

Parameters

Parameter	Description
hostName	Name of the host you want to delete. Enclose the host name in square brackets ([]). If the host name has more than one word, white spaces, or special characters, you must also enclose the host name in double quotes (“ ”).

Notes

A host is a computer that is attached to the storage array and accesses the volumes on the storage array through its host ports (host adapters).

Minimum Firmware Level

5.20

Delete Host Group

This command deletes a host group.

IMPORTANT This command will delete all host definitions in the host group.

Syntax

```
delete hostGroup [hostGroupName]
```

Parameters

Parameter	Description
<code>hostGroupName</code>	Name of the host group you want to delete. Enclose the host group name in square brackets ([]). If the host group name has more than one word, white spaces, or special characters, you must also enclose the host group name in double quotes (" ").

Notes

A host group is an optional topological element that is a collection of hosts that share access to the same volumes. The host group is a logical entity.

Minimum Firmware Level

5.20

Delete Host Port

This command deletes a host port.

Syntax

```
delete hostPort [hostPortName]
```

Parameters

Parameter	Description
hostPortName	Name of the host port you want to delete. Enclose the host port name in square brackets ([]).

Notes

A host port is a physical connection on a host adapter that resides within a host computer. A host port provides a host access to the volumes in a storage array. If the host adapter has only one physical connection (one host port), the terms host port and host adapter are synonymous.

Minimum Firmware Level

5.20

Delete Volume

This command deletes one or more standard volumes or snapshot and snapshot repository volumes.

CAUTION: Potential storage array configuration damage. All data in the volume is lost as soon as you run this command.

Syntax

```
delete (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN])
```

Parameters

Parameter	Description
<code>allVolumes</code>	Use this parameter to delete all volumes in a storage array.
<code>volumeName</code>	Name of the volume you want to delete. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Notes

When you use the `allVolumes` parameter, this command deletes volumes until all volumes are removed or until an error is encountered. If an error is encountered, this command will not attempt to delete the remaining volumes.

Minimum Firmware Level

6.10

Delete Volume Group

This command deletes an entire volume group and its associated volumes.

CAUTION: Potential storage array configuration damage. All data in the volume group is lost as soon as you run this command.

Syntax

```
delete volumeGroup [volumeGroupName]
```

.....

Parameters

Parameter	Description
volumeGroupNumber	Number of the volume group you want to delete. Enclose the volume group number in square brackets ([]).

Minimum Firmware Level

6.10

Diagnose Controller

This command runs diagnostic tests on the controller. The diagnostic tests consist of loop back tests in which data is written to physical disks and read from the physical disks.

Syntax

```
diagnose controller [(a | b)]  
loopbackDriveChannel=(allchannels | (1 | 2 | 3 | 4))  
testID=(1 | 2 | 3) [patternFile="filename"]
```

Parameters

Parameter	Description
(a b)	The controller on which you want to run the diagnostic tests. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]). If you do not specify a controller, the storage management software returns a syntax error.
loopbackDriveChannel	The physical disk channels on which you want to run the diagnostic tests. You can choose to run the diagnostics on “all channels” or select a specific channel on which to run diagnostics. If you want to select a specific channel, valid physical disk channel values are 1 , 2 , 3 , or 4 .

Parameter	Description
testID	The identification number of the diagnostic test you want to run. The identification numbers and corresponding tests are: 1 = Read Test 2 = Write Test 3 = Data Loop Back Test When you run a data loop back test you can optionally specify a file that contains a data pattern. If you do not specify a file, firmware provides a default pattern.
patternFile	The name of a file containing a data pattern that you want to use as test data. Enclose the data pattern filename in double quotes (“”).

Minimum Firmware Level

6.10

Diagnose Remote Mirror

This command tests the connection between the specified primary volumes and mirror volumes on a storage array with the Remote Volume Mirror feature installed.

Syntax

```
diagnose remoteMirror (primary [primaryVolumeName] |
primaries [primaryVolumeName1 ... primaryVolumeNameN])
testID=connectivity
```

Parameters

Parameter	Description
primaryVolumeName	Name of the primary volume of the Remote Volume Mirror you want to test. You can enter more than one primary volume name. Enclose the primary volume names in square brackets ([]). If the primary volume name has more than one word, white spaces, or special characters, you must also enclose the host group name in double quotes (“”).

.....

Minimum Firmware Level

6.10

Disable Storage Array Feature

This command disables a storage array feature. Issue the `show storageArray` command to display a list of the feature identifiers for all enabled features in the storage array.

Syntax

```
disable storageArray feature=(storagePartition2 |
storagePartition4 | storagePartition8 | storagePartition16 |
storagePartition64 | storagePartitionMax | snapshot |
remoteMirror | volumeCopy)
```

Parameters

None.

Notes

This command disables the Remote Volume Mirror feature and tears down the mirror repository volume.

Minimum Firmware Level

5.00

Download Drive Firmware

This command downloads a firmware image to a physical disk.

CAUTION **Potential storage array configuration damage.** Downloading physical disk firmware incorrectly can result in damage to the physical disks or loss of data.

Syntax

```
download drive [trayID,slotID]firmware file="filename"
```

Parameters

Parameter	Description
<code>trayID,slotID</code>	The physical disk to which you want to download the firmware image. Specify the tray ID and slot ID for the physical disk. Tray ID values are 0 to 99. Slot ID values are 1 to 32.
<code>filename</code>	The file path and filename of the file containing the firmware image. Enclose the firmware image file path and filename in double quotes (“ ”).

Notes

Before attempting to download physical disk firmware, you must take the following precautions:

- Stop all I/O activity to the storage array before downloading the firmware image.
- Ensure the firmware image file is compatible with the physical disk module. If you download a file that is not compatible with the physical disk module you have selected, the physical disk module may become unusable.
- Do not make any configuration changes to the storage array while downloading physical disk firmware. Attempting to make a configuration change can cause the firmware download to fail and make the selected physical disks unusable.

You can use this command to test the firmware on one physical disk before installing the firmware on all physical disks in a storage array. (Use the `download storageArray driveFirmware` command to download firmware on all physical disks in the storage array.) This command blocks all I/O activity until the download finishes or fails. The download returns one of the following statuses:

- Successful
- Unsuccessful with reason
- Never attempted with reason

Minimum Firmware Level

6.10

Download Environmental Card Firmware

This command downloads module firmware.

Syntax

```
download (allTrays | tray [trayID]) firmware file="filename"
```

Parameters

Parameter	Description
<code>trayID</code>	Identifies the module containing the environmental card to which you want to load new firmware. Tray ID values are 0 to 99. Enclose the tray ID value in square brackets ([]).
<code>filename</code>	The file path and filename of the file containing the firmware image. Enclose the firmware image file path and filename in double quotes (" ").

Notes

You can

- Use the `allTrays` parameter to download new firmware to all the modules in the storage array.
- Use the `tray` parameter to download new firmware to a specific module. If you need to download new firmware to more than one module, but not all modules, you will need to enter this command for each module.

Minimum Firmware Level

5.20

Download Storage Array Drive Firmware

This command downloads firmware images to all physical disks in the storage array.

Syntax

```
download storageArray driveFirmware file="filename"  
[file="filename2"...file="filenameN"]
```

Parameters

Parameter	Description
filename	The file path and filename of the file containing the firmware image. Enclose the firmware image file path and filename in double quotes (“”).

Notes

When you run this command, you can download more than one firmware image file to the physical disks in a storage array. The number of firmware image files you can download depends on the storage array. The storage management software will return an error if you attempt to download more firmware image files than the storage array can accept.

You can schedule downloads for multiple physical disks at the same time, including multiple physical disks in a redundant volume group. Each firmware image file contains information about the physical disk types on which the image will run. The specified firmware images will only be downloaded to a compatible physical disk. Use the `download drive firmware` command to download an image to a specific physical disk.

The `download storageArray driveFirmware` command blocks all I/O activity until a download attempt has been made for each candidate physical disk or you issue the `stop storageArray downloadDriveFirmware` command. When the `download storageArray driveFirmware` command finishes downloading the firmware image, each candidate physical disk is displayed showing the download status for each physical disk. Download status can be:

- Successful
- Unsuccessful with reason
- Never attempted with reason

Minimum Firmware Level

5.20

Download Storage Array Firmware/NVSRAM

This command downloads firmware and, optionally, NVSRAM values for the storage array controller. If you want to download only NVSRAM values, use the `download storageArray NVSRAM` command.

Syntax

```
download storageArray firmware [, NVSRAM ] file="filename"  
[, "NVSRAM-filename"] [downgrade=(TRUE|FALSE)]  
[activateNow=(TRUE|FALSE)]
```

Parameters

Parameter	Description
NVSRAM	Include this parameter when you want to download a file with NVSRAM values when you download a firmware file. Do not include square brackets with this parameter. Include a comma after the term <code>firmware</code> .
filename	The file path and name of the file containing the firmware. Valid filenames must end with a <code>.dlp</code> extension. Enclose the filename in double quotes (“ ”).
NVSRAM-filename	The file path and name of the file containing the NVSRAM values. Valid filenames must end with a <code>.dlp</code> extension. Enclose the NVSRAM filename in double quotes (“ ”). Include a comma after the firmware filename.
downgrade	Use this parameter if you are loading firmware that is a previous version. The default is FALSE . Set <code>downgrade</code> to TRUE if you want to download an earlier version of firmware.
activateNow	Use this parameter to activate the firmware and NVSRAM images. The default is TRUE . If you set <code>activateNow</code> to FALSE , you must use the <code>activate storageArray firmware</code> command to activate the firmware and NVSRAM values at a later time.

Minimum Firmware Level

5.00

Download Storage Array NVSRAM

This command downloads NVSRAM values for the storage array controller.

Syntax

```
download storageArray NVSRAM file="filename"
```

Parameters

Parameter	Description
filename	The file path and name of the file containing the NVSRAM values. Valid filenames must end with a .dlp extension. Enclose the filename in double quotes (“ ”).

Minimum Firmware Level

6.10

Enable Controller

This command revives a controller that has become quiesced while running diagnostics.

Syntax

```
enable controller [(a | b)] dataTransfer
```

.....

Parameters

Parameter	Description
(a b)	The controller that you want to revive. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]). If you do not specify a controller, the storage management software returns a syntax error.

Minimum Firmware Level

6.10

Enable Storage Array Feature Key

This command enables a feature using a feature key file.

Syntax

```
enable storageArray feature file="filename"
```

Parameters

Parameter	Description
filename	The file path and filename of a valid feature key file. Enclose the file path and filename in double quotes (" "). Valid filenames for feature key files end with a .key extension.

Minimum Firmware Level

6.10

Recopy Volume Copy

This command reinitiates a volume copy operation using an existing volume copy pair.

Syntax

```
recopy volumeCopy target [targetName] [source [sourceName]]  
[copyPriority=(highest | high | medium | low | lowest)]  
targetReadOnlyEnabled=(TRUE | FALSE)]
```

Parameters

Parameter	Description
target	The name of the target volume for which you want to reinitiate a volume copy operation. Enclose the target volume name in square brackets ([]). If the target volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
source	The name of the source volume for which you want to reinitiate a volume copy operation. Enclose the source volume name in square brackets ([]). If the source volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
copyPriority	Defines the priority volume copy will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest
targetReadOnlyEnabled	Defines whether you can write to the target volume or only read from the target volume. To be able to write to the target volume, set this parameter to FALSE . To prevent writing to the target volume, set this parameter to TRUE .

Notes

Copy priority defines the amount of system resources used to copy the data between the source and target volumes of a volume copy pair. If you select the highest priority level, the volume copy uses the most system resource to perform volume copy, which will decrease performance for host data transfers.

Minimum Firmware Level

6.10

Recover RAID Volume

This command creates a RAID volume with the given properties without initializing any of the user data areas on the disks. Parameter values are derived from the Recovery Profile data file for the storage array.

Syntax

```
recover volume (drive=(trayID,slotID) |  
drives=(trayID1,slotID1 ... trayIDn,slotIDn) |  
volumeGroup=volumeGroupNumber) userLabel="volumeName"  
capacity=volumeCapacity offset=offsetValue raidLevel=(0 | 1  
| 3 | 5) segmentSize=segmentSizeValue [owner=(a | b)  
readAheadMultiplier=multiplierValue]
```

Parameters

Parameter	Description
drive	Defines the physical disks you want to assign to the volume you want to create. Specify the tray ID and slot ID for each physical disk you assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in parentheses.
volumeGroup	The sequence number of the volume group in which you want to create the new volume. (To determine the sequence numbers of the volume groups in your storage array, enter the <code>show storageArray profile</code> command.)
userLabel	The name you want to give the new volume. Enclose the new volume name in double quotes (“”).
capacity	The size of the volume you are adding to the storage array. Size is defined in units of bytes, KB, MB, or GB.

Parameter	Description
offset	The number of blocks from the beginning of the volume group to the beginning of the referenced volume.
raidLevel	The RAID level of the volume group that contains the physical disks. Valid RAID levels are 0 , 1 , 3 , or 5 .
segmentSize	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.
owner	The controller that owns the volume. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
readAheadMultiplier	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfers.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

The `owner` parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the volume group.

Cache read-ahead enables the controller to copy additional data blocks into cache while the controller reads and copies host requested data blocks from disk into cache. This

.....

increases the chance that a future request for data can be fulfilled from cache. Cache read-ahead is important for multimedia applications that use sequential data transfers. Choosing a higher cache read-ahead multiplier might increase the cache hit percentage, however, transfer time might also increase.

Minimum Firmware Level

5.43

Re-create Remote Volume Mirroring Repository

This command creates a new Remote Volume Mirror repository volume using the parameters defined for a previous Remote Volume Mirror repository volume. The underlying requirement is that you have previously created a Remote Volume Mirror repository volume. When you use this command, you can define the repository volume in one of three ways: user-defined physical disks, user-defined volume group, or user-defined number of physical disks for the repository volume. If you choose to define a number of physical disks, the controller firmware chooses which physical disks to use for the repository volume.

Syntax (User-Defined Physical Disks)

```
recreate storageArray mirrorRepository
repositoryRAIDLevel=(1 | 3 | 5)
repositoryDrives=(trayID1,slotID1 ... trayIDn,slotIDn
[trayLossProtect=(TRUE | FALSE)])
```

Syntax (User-Defined Volume Group)

```
recreate storageArray mirrorRepository
repositoryRAIDLevel=(1 | 3 | 5)
repositoryVolumeGroup=volumeGroupNumber
[repositoryFreeCapacityArea=freeCapacityValue
trayLossProtect=(TRUE | FALSE)]
```

Syntax (User-Defined Number of Physical Disks)

```
recreate storageArray mirrorRepository
repositoryRAIDLevel=(1 | 3 | 5) repositoryDriveCount=
numberOfDrives [driveType=(fibre | SATA | PATA)
trayLossProtect=(TRUE | FALSE)]
```

Parameters

Parameter	Description
<code>repositoryRAIDLevel</code>	The RAID level for the repository volume. Valid values are 1 , 3 , or 5 .
<code>repositoryDrives</code>	The physical disks for the repository volume. Specify the tray ID and slot ID for each physical disk you assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in parentheses.
<code>repositoryVolumeGroup</code>	The sequence number of the volume group where the repository volume is located.
<code>repositoryFreeCapacityArea</code>	The free space in an existing volume group that you want to use to create the mirror repository volume. Free capacity is defined as the free capacity between existing volumes in a volume group. For example, a volume group might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3 free capacity. To use the free capacity following volume 2, you would specify: <code>repositoryFreeCapacityArea=2.</code>
<code>repositoryDriveCount</code>	The number of unassigned physical disks that you want to use for the repository volume.
<code>driveType</code>	The type of physical disks that you want to use for the repository volume. Valid physical disk types are: Fibre , SATA , or PATA .
<code>trayLossProtect</code>	Specifies that tray loss protection will be enforced when creating the repository. To enforce tray loss protection, set this parameter to TRUE . The default setting is FALSE .

Notes

If you enter a value for the repository storage space that is too small for the mirror repository volumes, the firmware will return an error message giving the amount of space needed for the mirror repository volumes. The command will not attempt to change the Remote Volume Mirror repository. You can re-enter the command using the value from the error message for the repository storage space value.

When you assign the physical disks, if you set `trayLossProtect` to true and have selected more than one physical disk from any one module, the storage array will return an error. If you set `trayLossProtect` to false, the storage array will perform operations, but the volume group you create may not have tray loss protection.

When the controller firmware assigns the physical disks, if you set `trayLossProtect` to true the storage array will error the function if the controller firmware cannot provide drives that will result in the new volume group having tray loss protection. If you set `trayLossProtect` to false the storage array will perform the operation even if it means the volume group may not have tray loss protections.

Minimum Firmware Level

6.10

Re-create Snapshot

This command starts a fresh copy-on-write operation using an existing snapshot volume.

Syntax

```
recreate snapshot (volume [volumeName] | volumes
[volumeName1 ... volumeNameN])
[userLabel= "snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryFullPolicy=(failBaseWrites | failSnapShot)]
```

Parameters

Parameter	Description
<code>volumeName</code>	The name of the specific volume for which you want to start a fresh copy-on-write operation. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (" ").
<code>userLabel</code>	The name of the snapshot volume. Enclose the snapshot volume name in double quotes (" "). If you enter more than one snapshot volume name this command will fail.

Parameter	Description
<code>warningThresholdPercent</code>	The percentage of repository capacity at which you will receive a warning that the repository is nearing full. Use integer values. For example, a value of 70 means 70 percent. The default value is 50 percent.
<code>repositoryFullPolicy</code>	Defines how you want snapshot processing to continue if the repository is full. You can choose to fail writes to the base volume (<code>failBaseWrites</code>) or fail writes to the snapshot volume (<code>failSnapshot</code>). The default is <code>failSnapshot</code> .

Notes

You can use any combination of alphanumeric characters and the underscore (`_`) and dash (`-`) characters for the names.

If `warningThresholdPercent` or `repositoryFullPolicy` are not specified, the previously set values are used.

Minimum Firmware Level

5.00

Remove Remote Volume Mirror

This command removes the mirror relationship between the primary volume and secondary volume.

Syntax

```
remove remoteMirror (localVolume [volumeName] | localVolumes
[volumeName1 ... volumeNameN])
```

.....

Parameters

Parameter	Description
<code>localVolume</code>	Name of the primary volume (volume on the local storage array) you want to remove. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Minimum Firmware Level

6.10

Remove Volume Copy

This command removes a volume copy pair.

Syntax

```
remove volumeCopy target [targetName] [source [sourceName]]
```

Parameters

Parameter	Description
<code>target</code>	The name of the target volume you want to remove. Enclose the target volume name in square brackets ([]). If the target volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
<code>source</code>	The name of the source volume you want to remove. Enclose the source volume name in square brackets ([]). If the source volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Minimum Firmware Level

5.40

Remove Volume LUN Mapping

This command removes the logical unit number mapping.

Syntax

```
remove (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN] | accessVolume) lunMapping  
(host= "hostName" | hostGroup= ("hostGroupName" |  
defaultGroup))
```

Parameters

Parameter	Description
allVolumes	Use this parameter to remove the LUN mapping from all volumes.
volumeName	Name of the specific volume that you want to remove from the LUN mapping. You can enter more than one volume name. Enclose the volume name in double quotes (“ ”) inside square brackets ([]).
accessVolume	Use this parameter to remove the access volume.
host	The name of the host to which the volume is mapped. Enclose the host name in double quotes (“ ”).
hostGroup	The name of the host group that contains the host to which the volume is mapped. Enclose the host group name in double quotes (“ ”). defaultGroup is the host group that contains the host to which the volume is mapped.

Notes

The access volume is the volume in a SAN environment used for communication between the storage management software and the storage array controller. This volume uses a LUN address and consumes 20 MB of storage space that is not available for application data storage. An access volume is required only for in-band managed storage arrays.

CAUTION **Removing an access volume can damage your configuration.** The agent uses the access volumes to communicate with a storage array. If you remove an access volume mapping for a storage array from a host that has an agent running on it, the storage management software will no longer be able to manage the storage array through the agent.

You must use the `host` and `hostGroup` parameters when specifying a non-access volume or an access volume. The script engine ignores the `host` or `hostGroup` parameters when you use the `allVolumes` or `volumes` parameters.

Minimum Firmware Level

6.10

Repair Volume Parity

This command repairs the parity errors on a volume.

Syntax

```
repair volume [volumeName] parity parityErrorFile=filename
[verbose=(TRUE | FALSE)]
```

Parameters

Parameter	Description
<code>volumeName</code>	The name of the specific volume for which you want to repair parity. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the snapshot volume name in double quotes (“ ”).
<code>parityErrorFile</code>	The name of the file that contains the parity error information you will use to repair the errors. Enclose the filename in double quotes (“ ”).
<code>verbose</code>	Captures progress details, such as percent complete, and displays the information as volume parity is being repaired. To capture progress details, set this parameter to TRUE . To prevent capturing progress details, set this parameter to FALSE .

Minimum Firmware Level

6.10

Reset Controller

This command resets a controller.

IMPORTANT When you reset a controller, the controller is not available for I/O operations until the reset is complete. If a host is using volumes owned by the controller being reset, the I/O directed to the controller will be rejected. Before resetting the controller, either verify that the volumes owned by the controller are not in use or ensure a multi-path driver is installed on all hosts using these volumes.

Syntax

```
reset controller [(a | b)]
```

Parameters

Parameter	Description
(a b)	The controller that you want to reset. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]). If you do not specify an owner, the controller firmware returns a syntax error.

Minimum Firmware Level

5.20

Reset Storage Array Battery Install Date

This command resets the age of the batteries in a storage array to zero days. You can reset the batteries for an entire storage array or the battery in a specific controller.

Syntax

```
reset storageArray batteryInstallDate [controller=(a | b)]
```

Parameters

Parameter	Description
<code>controller</code>	Defines the controller that contains the battery for which you want to reset the age. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B.

Notes

If you do not specify a controller, the age for the storage array battery or both controller batteries is reset. If you specify a controller, then the age for only that controller battery is reset. If the battery is housed in a container separate from the controllers, the `controller` parameter is ignored.

Minimum Firmware Level

6.10

Reset Storage Array RLS Baseline

This command resets the Read Link Status (RLS) baseline for all devices.

Syntax

```
reset storageArray RLSBaseline
```

Parameters

None.

Minimum Firmware Level

5.00

Reset Storage Array Volume Distribution

This command reassigns (moves) all volumes to their “preferred” controller.

Syntax

```
reset storageArray volumeDistribution
```

Parameters

None.

Notes

IMPORTANT If you use this command on a host without a multi-path driver, to prevent application errors you must stop I/O operations to the volumes until this command is completed.

IMPORTANT Under certain host operating system environments, you may be required to reconfigure the multi-path host physical disk. You may also need to make operating system modifications to recognize the new I/O path to the volumes.

Minimum Firmware Level

5.20

Resume Remote Volume Mirror

This command resumes a suspended Remote Volume Mirror operation.

Syntax

```
resume remoteMirror (primary [volumeName] | primaries  
[volumeName1 ... volumeNameN]) writeConsistency=(TRUE |  
FALSE)
```

Parameters

Parameter	Description
<code>primary</code>	Name of the primary volume for which you want to resume operation. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
<code>writeConsistency</code>	Defines whether the volumes identified in this command must be in a write-consistency group or can be separate. For the volumes to be in the same write-consistency group, set this parameter to TRUE . For the volumes to be separate, set this parameter to FALSE .

Notes

If you set `writeConsistency` to `TRUE`, the volumes must be in a write-consistency group (or groups). The action of the command is to resume all consistency groups containing the volumes. For example, if volumes A, B, and C are in a consistency group and they have remote counterparts A', B' and C', the command `resume remoteMirror volume ["A"] writeConsistency=true` resumes A-A', B-B', and C-C'.

Minimum Firmware Level

6.10

Revive Drive

This command forces the specified physical disk to the optimal state.

Syntax

```
revive drive [trayID,slotID]
```

Parameters

Parameter	Description
<code>trayID,slotID</code>	Identifies the module and slot where the physical disk resides. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).

Minimum Firmware Level

5.43

Revive Volume Group

This command forces the specified volume group and associated failed physical disks to the optimal state.

Syntax

```
revive volumeGroup [volumeGroupName]
```

Parameters

Parameter	Description
<code>volumeGroupName</code>	Number of the volume group to be set to the optimal state. Enclose the volume group number in square brackets ([]).

Minimum Firmware Level

6.10

Save Controller NVSRAM

This command saves a copy of the controller NVSRAM values to a file. This command saves all regions.

.....

Syntax

```
save controller [(a | b)] NVSRAM file="filename"
```

Parameters

Parameter	Description
(a b)	The controller with the NVSRAM values you want to save. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]).
filename	The name of the file in which you want to save the values. Enclose the filename in double quotes ("").

Minimum Firmware Level

6.10

Save Drive Log

This command saves the log sense data to a file. Log sense data is maintained by the storage array for each physical disk.

Syntax

```
save allDrives logFile="filename"
```

Parameters

Parameter	Description
filename	Name of the file to which you want to write the sense log data. Enclose the filename in double quotes ("").

Minimum Firmware Level

6.10

Save Storage Array Configuration

This command creates a script file that you can use to create the current storage array volume configuration.

Syntax

```
save storageArray configuration file="filename" [(allconfig  
| globalSettings=(TRUE | FALSE)  
volumeConfigAndSettings=(TRUE | FALSE) hostTopology=(TRUE |  
FALSE) lunMappings=(TRUE | FALSE))]
```

Parameters

Parameter	Description
filename	The name of the file that will contain the configuration values. Enclose the filename in double quotes (“”).
allConfig	Saves all the configuration values to the file. (If you choose this parameter all configuration parameters are set to true.)
globalSettings	Saves the global settings to the file. To save the global settings, set this parameter to TRUE . To prevent saving the global settings, set this parameter to FALSE . The default is TRUE .
volumeConfigAndSettings	Saves the volume configuration settings and all global settings to the file. To save the volume configuration and global settings, set this parameter to TRUE . To prevent saving the volume configuration and global settings, set this parameter to FALSE . The default is TRUE .
hostTopology	Saves the host topology to the file. To save the host topology, set this parameter to TRUE . To prevent saving the host topology, set this parameter to FALSE . The default is FALSE .
lunMapping	Saves the LUN mapping to the file. To save the LUN mapping, set this parameter to TRUE . To prevent saving the LUN mapping, set this parameter to FALSE . The default is FALSE .

Notes

When you use this command, you can specify any combination of the parameters for global setting, volume configuration setting, host topology, or LUN mapping. If you want to enter all settings, use the `allConfig` parameter. The parameters are optional; you do not have to enter any parameters.

Minimum Firmware Level

6.10

Save Storage Array Events

This command saves events from the major event log (MEL) to a file. You can save either all the events or only the critical events.

Syntax

```
save storageArray (allEvents | criticalEvents)
file="filename" [count=numberOfEvents]
```

Parameters

Parameter	Description
<code>allEvents criticalEvents</code>	Defines whether you want to save all events or only the critical events.
<code>filename</code>	The name of the file to which you will save the events. Enclose the filename in double quotes (“ ”).
<code>count</code>	Specifies the number of events or critical events you want to save to a file. If you do not enter a value for the count, all events or critical events are saved to the file. If you enter a value for the count, only that number of events or critical events (starting with the last event entered) are saved to the file. Use integer values.

Minimum Firmware Level

6.10

Save Storage Array Performance Statistics

This command saves the performance statistics to a file. Before you use this command, issue the `set session performanceMonitorInterval` and `set session performanceMonitorIterations` commands to specify how often statistics are collected.

Syntax

```
save storageArray performanceStats file="filename"
```

Parameters

Parameter	Description
filename	The name of the file to which you will save the performance statistics. Enclose the filename in double quotes (“ ”).

Minimum Firmware Level

6.10

Save Storage Array RLS Counts

This command saves the RLS counters to a file. Before using this command, issue the `reset storageArray RLSBaseline` command to get current data.

Syntax

```
save storageArray RLSCounts file="filename"
```

Parameters

Parameter	Description
filename	The name of the file to which you will save the RLS counters. Enclose the filename in double quotes (“ ”).

Minimum Firmware Level

6.10

Save Storage Array State Capture

This command saves the state capture to a file.

Syntax

```
save storageArray stateCapture file="filename"
```

Parameters

Parameter	Description
filename	The name of the file to which you will save the state capture. Enclose the filename in double quotes (“”).

Minimum Firmware Level

6.10

Save Storage Array Support Data

This command saves the support related information to a file. Support related information includes:

- Storage array profile
- Major event log (MEL) information
- Read link status (RLS) data
- NVSRAM data
- Current problems and associated recovery information
- Performance statistics for the entire storage array
- Persistent registration and reservation information
- Detailed information about the current status of the storage array
- Physical disk diagnostic data

- A recovery profile for the storage array
- Unreadable sectors detected on the storage array
- State capture data

Syntax

```
save storageArray supportData file="filename"
```

Parameters

Parameter	Description
filename	The name of the file to which you will save the support related data. Enclose the filename in double quotes (“ ”).

Minimum Firmware Level

6.10

Set Controller

This command defines the properties for the controllers.

Syntax

```
set controller [(a | b)] availability=(online | offline |
serviceMode) bootp gatewayIPAddress=ipAddress
globalNVSramByte [nvsramOffset]=(nvsramByteSetting |
nvsramBitSetting) hostNVSramByte [hostType,
nvsramOffset]=(nvsramByteSetting | nvsramBitSetting)
ipAddress=ipAddress rloginEnabled=(TRUE | FALSE)
subnetMask=ipAddress
```

.....

Parameters

Parameter	Description
(a b)	The controller for which you want to define properties. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]). If you do not specify a controller, the controller firmware returns a syntax error.
availability	Sets the controller mode to online, offline, or in service.
bootp	Causes a controller to find a bootp server. The bootp server will then assign the controller an IP address.
gatewayIPAddress	Defines the gateway IP address for the controller. Setting the gateway IP address to 0.0.0.0 will enable bootp.
globalNVSRAMByte	Modifies a portion of the controller NVSRAM. Specify the region to be modified using the starting byte offset within the region, and the size and value of the new data to be stored into NVSRAM.
hostNVSRAMByte	Updates the NVSRAM for the host specific region. Specifies the host index for the specific host, the starting offset within the region, the number of bytes, and the values to be written.
nvsramoffset	Starting byte for the globalNVSRAMByte and hostNVSRAMByte parameter
nvsramByteSetting	New byte value when you are changing NVSRAM values.
nvsramBitSetting	New bite value when you are changing NVSRAM values.
ipAddress	Controller IP address.
rloginEnabled	Defines whether the remote login feature is turned on or turned off. To turn on the remote login feature, set this parameter to TRUE . To turn off the remote login feature, set this parameter to FALSE .
subnetMask	Controller subnet mask.

Notes

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

Use the `show controller NVSRAM` command to display parts or all of the NVSRAM.

Minimum Firmware Level

`gatewayIPAddress` parameter 5.40

`ipAddress` parameter 5.40

`rloginEnabled` parameter 5.40

`subnetMask` parameter 5.40

`serviceMode` parameter 5.43

Set Drive Channel Status

This command defines how the physical disk channel performs.

Syntax

```
set driveChannel [(1 | 2 | 3 | 4)] status=(optimal | degraded)
```

Parameters

Parameter	Description
<code>driveChannel</code>	The identifier number of the physical disk channel for which you want to set the status. Valid physical disk channel values are 1 , 2 , 3 , or 4 . Enclose the physical disk channel number in square brackets ([]).
<code>status</code>	The condition of the channel. You can set the channel status to optimal or degraded .

Notes

Use the optimal parameter to move a degraded channel back to optimal. Use the degraded parameter when the channel is experiencing problems and the storage array requires additional time for data transfers.

Minimum Firmware Level

6.10

Set Drive Hot Spare

This command assigns or unassigns one or more physical disks as a hot spare.

Syntax

```
set (drive [trayID,slotID] | drives [trayID1,slotID1 ...  
trayIDn,slotIDn]) hotSpare=(TRUE | FALSE)
```

Parameters

Parameter	Description
<i>trayID</i> , <i>slotID</i>	Identifies the module and slot where the physical disk resides. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).
hotSpare	Assigns the physical disk as the hot spare. To assign the physical disk as the hot spare, set this parameter to TRUE . To remove a hot spare assignment from a physical disk, set this parameter to FALSE .

Minimum Firmware Level

6.10

Set Drive State

This command sets a physical disk to the failed state. (To return a physical disk to the optimal state, use the `revive drive` command.)

Syntax

```
set drive [trayID,slotID] operationalState=failed
```

Parameters

Parameter	Description
<code>trayID,slotID</code>	Identifies the module and slot where the physical disk resides. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).

Minimum Firmware Level

5.20

Set Host

This command assigns a host to a host group or moves a host to a different host group. You can also create a new host group and assign the host to the new host group with this command. The actions performed by this command depend on whether the host has individual volume-to-LUN mappings or does not have individual volume-to-LUN mappings.

Syntax

```
set host [hostName] hostGroup=( "hostGroupName" | none |  
defaultGroup) userLabel= "newHostName"
```

Parameters

Parameter	Description
<code>hostName</code>	The name of the host that you want to assign to a host group. Enclose the host name in square brackets ([]). If the host name has more than one word, white spaces, or special characters, you must also enclose the host name in double quotes (" ").

.....

Parameter	Description
hostGroup	The host group to which you want to assign the host. (The following table defines how the command runs if the host does or does not have individual volume-to-LUN mappings.) Enclose the host group name in double quotes (“ ”). defaultGroup is the host group that contains the host to which the volume is mapped.
userLabel	New host name. Enclose the host name in double quotes (“ ”).

Host Group Parameter	Host Has Individual Volume to LUN Mappings	Host Does Not Have Individual Volume to LUN Mappings
hostGroupName	Host is removed from the present host group and is placed under the new host group defined by hostGroupName.	Host is removed from the present host group and is placed under the new host group defined by hostGroupName.
none	Host is removed from the host group as an independent partition and is placed under the root node.	Host is removed from the present host group and is placed under the default group.
defaultGroup	Command fails.	Host is removed from the present host group and is placed under the default group.

Notes

When you use this command, you can specify one or more of the optional parameters. You do not, however, need to use all of the parameters.

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

6.10

Set Host Channel

This command defines the loop ID for the host channel.

Syntax

```
set hostChannel [(a1 | a2 | b1 | b2)] preferredID=portID
```

Parameters

Parameter	Description
hostChannel	The identifier number of the host channel for which you want to set the loop ID. Enclose the host channel identifier number in square brackets ([]).
portID	The new port ID for the specified host channel. Port ID values are 0-127.

Minimum Firmware Level

6.10

Set Host Group

This command renames a host group.

Syntax

```
set hostGroup [hostGroupName] userLabel="newHostGroupName"
```

Parameters

Parameter	Description
hostGroupName	The name of the host group that you want to rename. Enclose the host group name in square brackets ([]). If the host group name has more than one word, white spaces, or special characters, you must also enclose the host group name in double quotes (" ").

Parameter	Description
<code>newHostGroupName</code>	The new name for the host group. Enclose the new host group name in double quotes (“ ”).

Notes

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

6.10

Set Host Port

This command changes the host type for a host port. You can also change a host port label with this command.

Syntax

```
set hostPort [portLabel] host="hostName"
hostType=(hostTypeIndexLabel | hostTypeIndexNumber)
userLabel="newPortLabel"
```

Parameters

Parameter	Description
<code>portLabel</code>	Name of the host port for which you want to change the host type, or for which you want to create a new name. Enclose the host port name in square brackets ([]). If the host port label has more than one word, white spaces, or special characters, enclose the host port label in double quotes (“ ”).
<code>host</code>	The name of the host to which the host port is or will be connected. Enclose the host name in double quotes (“ ”).

Parameter	Description
hostType	The index label or number of the host port host type. Use the <code>show storageArray hostTypeTable</code> command to generate a list of available host type identifiers. If the host type has more than one word, white spaces, or special characters, enclose the host type in double quotes (“ ”).
userLabel	New name that you want to give to the host port. Enclose the new host port name in double quotes (“ ”).

Notes

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

You can use any combination of alphanumeric characters and the underscore (_) and dash (-) characters for the names.

Minimum Firmware Level

6.10

Set Remote Volume Mirror

This command defines the properties for a Remote Volume Mirror pair.

Syntax

```
set remoteMirror (localVolume [volumeName] | localVolumes
[volumeName1 ... volumeNameN]) role=(primary | secondary)
[force=(TRUE | FALSE)] syncPriority=(highest | high | medium
| low | lowest) writeOrder=(preserved | notPreserved)
writeMode=(synchronous | asynchronous)
```

Parameters

Parameter	Description
volumeName	Name of the primary volume for which you want to define properties. You can enter more than one primary volume name. Enclose the primary volume name in square brackets ([]). If the primary volume name has more than one word, white spaces, or special characters, you must also enclose the primary volume name in double quotes (“ ”).
role	Defines whether the volume acts as the primary or secondary volume. To define the volume as the primary volume, set this parameter to primary . To define the volume as the secondary volume, set this parameter to secondary . This parameter applies only when the the volume is part of a mirror relationship.
force	Defines that you want to force the role reversal if the communications link between the storage arrays is down and promotion or demotion on the local side will result in a dual primary or dual secondary condition. To force a role change, set this parameter to TRUE . The default is FALSE .
syncPriority	Defines the priority full synchronization will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest
writeOrder	Defines write order for data transmission between the primary volume and secondary volume. Valid values are preserved or notPreserved .
writeMode	Defines how the primary volume writes to the secondary volume. Valid values are synchronous or asynchronous .

Notes

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

.....

Synchronization priority defines the amount of system resources used to synchronize the data between the primary and secondary volumes of a mirror relationship. If you select the highest priority level, the data synchronization uses the most system resource to perform the full synchronization, which will decrease performance for host data transfers.

Setting the writer order to **preserved** causes the mirrored pair to transmit data from the primary to secondary in the same order as the host writes to the primary. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This can require additional system overhead to maintain the buffered data, slowing operations. Setting write order to **notPreserved** frees the system from having to maintain data in a buffer, but requires forcing a full synchronization to ensure the secondary volume has the same data as the primary.

Minimum Firmware Level

6.10

Set Session

This command defines how you want the current script engine session to run.

Syntax

```
set session errorAction=(stop | continue)
password="storageArrayPassword"
performanceMonitorInterval=intervalValue
performanceMonitorIterations=iterationValue
```

Parameters

Parameter	Description
<code>errorAction</code>	Defines how the session will respond if an error is encountered during processing. You can choose to stop the session if an error is encountered or you can continue after encountering an error. The default error action is to stop. (This parameter defines the action for execution errors, not syntax errors. Some error conditions may over ride the continue value.)
<code>password</code>	The password for the storage array. Enclose the password in double quotes (“”).

Parameter	Description
<code>performanceMonitorInterval</code>	Defines how frequently to gather performance data. Enter an integer value for the polling interval, in seconds, for which you want to capture data. The range of values is 3 to 3600 seconds. The default value is 5 seconds.
<code>performanceMonitorIterations</code>	Defines the number of samples to capture. Enter an integer value. The range of values for samples captured is 1 to 3600. The default value is 5.

Notes

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of alpha-numeric characters with a maximum of 30 characters. (You can define a storage array password using the `set storageArray` command.)

The polling interval and number of iterations you specify remain in effect until you end the session. After you end the session, the polling interval and number of iterations return to the default values.

Minimum Firmware Level

5.20

Set Snapshot Volume

This command defines the properties for a snapshot volume and enables you to rename a snapshot volume.

Syntax

```
set (volume [volumeName] | volumes [volumeName1 ...
volumeNameN]) userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryFullPolicy=(failBaseWrites | failSnapShot)
```

Parameters

Parameter	Description
<code>volumeName</code>	The name of the specific snapshot volume for which you want to define properties. (You can enter more than one volume name if you use the <code>volumeNames</code> parameter). Enclose the snapshot volume name in double quotes (“ ”) inside square brackets ([]).
<code>userLabel</code>	A new name you want to give to a snapshot volume. Enclose the new snapshot volume name in double quotes (“ ”).
<code>warningThresholdPercent</code>	The percentage of repository capacity at which you will receive a warning that the repository is nearing full. Use integer values. For example, a value of 70 means 70 percent. The default value is 50 percent.
<code>repositoryFullPolicy</code>	Defines how you want snapshot processing to continue if the repository is full. You can choose to fail writes to the base volume (<code>failBaseWrites</code>) or fail writes to the snapshot volume (<code>failSnapShot</code>). The default is <code>failSnapShot</code> .

Notes

When you use this command, you can specify one or more of the optional parameters. You do not, however, need to use all of the parameters.

You can use any combination of alphanumeric characters and the underscore (`_`) and dash (`-`) characters for the names.

You can set the `warningThresholdPercent` and the `repositoryFullPolicy` parameters for both the repository volume or snapshot volume.

Minimum Firmware Level

6.10

Set Storage Array

This command defines the properties of the storage array.

Syntax

```
set storageArray cacheBlockSize=cacheBlockSizeValue
cacheFlushStart=cacheFlushStartSize
cacheFlushStop=cacheFlushStopSize
defaultHostType=(hostTypeName | hostTypeIdentifier)
failoverAlertDelay=delayValue mediaScanRate=(disabled | 1-
30) password=password userLabel=storageArrayName
```

Parameters

Parameter	Description
cacheBlockSize	Defines the cache block size used by the controller for managing the cache. Valid values are 4 KB or 16 KB.
cacheFlushStart	Defines the percentage of unwritten data in the cache that will cause a cache flush. Use integer values from 0 to 100 to define the percentage.
cacheFlushStop	Defines the percentage of unwritten data in the cache that will stop a cache flush in progress. Use integer values from 0 to 100 to define the percentage.
defaultHostType	Defines the default host type of any unconfigured host port to which the controllers are connected. To generate a list of valid host types for the storage array enter the <code>show storageArray hostTypeTable</code> command. Host types can be identified by a name or numerical index. Enclose the host type name in double quotes (“”). Do not enclose the host type numerical identifier in double quotes.
failoverAlertDelay	Defines the failover alert delay time in minutes. The valid delay time range is 0 to 60 minutes. The default is 5 minutes.
mediaScanRate	Defines the number of days over which the media scan runs. Valid values are: disabled , which turns off media scan, or 1-30 , where 1 is the fastest scan rate and 30 is the slowest. A value of 0 will disable media scan.

Parameter	Description
password	Defines the password for the storage array. Enclose the password in double quotes (“ ”).
userLabel	Defines a name for the storage array. Enclose the storage array name in double quotes (“ ”).

Notes

When you use this command, you can specify one or more of the optional parameters. You do not, however, need to use all of the parameters.

When defining cache block sizes, use the 4 KB cache block size for systems that require I/O streams that are typically small and random. Use the 16 KB cache block size for systems that require large data transfer, sequential, high bandwidth applications. This parameter defines the cache block size for all volumes in the storage array. For redundant configurations, this includes all volumes owned by both controllers within the storage array.

When defining values to start a cache flush, a value that is too low increases the chance that data needed for a host read will not be in the cache. It also increases the number of disk writes necessary to maintain the cache level, which increases system overhead and decreases performance. If you enter a value the controller cannot support, the controller will return a list of appropriate values.

When defining values to stop a cache flush, the lower the value the higher the chance that data for a host read will require a disk read rather than reading from the cache.

When defining host types, if SANshare storage partitioning is enabled, the default host type affects only those volumes mapped in the default group. If SANshare storage partitioning is not enabled, all hosts attached to the storage array must run the same operating system and be compatible with the default host type.

Media scan runs on all volumes in the storage array that are optimal, have no modification operations in progress, and have the media scan parameter enabled.

Passwords are stored on each storage array. The password can be any combination of alpha-numeric characters with a maximum of 30 characters.

Minimum Firmware Level

defaultHostType parameter 5.00

failoverAlertDelay parameter 5.40

Set Storage Array Time

This command sets the clocks on both controllers in a storage array by synchronizing the controller clocks with the clock of the host from which you issue this command.

Syntax

```
set storageArray time
```

Parameters

None.

Minimum Firmware Level

6.10

Set Storage Array Tray Positions

This command defines the position of all modules in the storage array. All modules in the storage array must be included in the definition list.

Syntax

```
set storageArray trayPositions=(controller | 0 - 99)
```

Parameters

Parameter	Description
trayPositions	The parameters for this command are the values for the module position. Valid values are 0 to 99. Enter the module position values separated by a space. Enclose the list of module values in parentheses. For storage arrays where the command module has a predefined identifier that is not in the range of valid module position values, use the term <code>controller</code> value.

Minimum Firmware Level

6.10

Set Volume

This command defines the properties for a volume. You can use most parameters to define properties for one or more volumes; however, you can use some parameters to define properties for only one volume. The syntax definitions are separated to show which parameters apply to several volumes and which apply to only one volume. Also, the syntax for volume mapping is listed separately.

Syntax Applicable to One or More Volumes

```
set (allVolumes | volume [volumeName] | volumes [volumeName1
... volumeNameN] | volume <wwn>)
cacheFlushModifier=cacheFlushModifierValue
cacheWithoutBatteryEnabled=(TRUE | FALSE)
mediaScanEnabled=(TRUE | FALSE) mirrorCacheEnabled=(TRUE |
FALSE) modificationPriority=(highest | high | medium | low |
lowest) owner=(a | b) readCacheEnabled=(TRUE | FALSE)
writeCacheEnabled=(TRUE | FALSE)
readAheadMultiplier=integer-literal
```

Syntax Applicable to Only One Volume

```
set (volume [volumeName] | volume <wwn>)
addCapacity=volumeCapacity [addDrives=(trayID1,slotID1 ...
trayIDn,slotIDn)] redundancyCheckEnabled=(TRUE | FALSE)
segmentSize=segmentSizeValue userLabel=volumeName
```

Syntax Applicable to Volume Mapping

```
set (volume [volumeName] | volume <wwn> | accessVolume)
logicalUnitNumber=LUN (host="hostName" |
hostGroup=("hostGroupName" | defaultGroup))
```

Parameters

Parameter	Description
allVolumes	Use this parameter to define the properties for all volumes in the storage array.

Parameter	Description
volumeName	The name of the specific volume for which you want to define properties. (You can enter more than one volume name if you use the volumeNames parameter). Enclose the snapshot volume name in double quotes (“ ”) inside square brackets ([]).
wwn	The world wide name of the volume for which you are setting properties. You can use the world wide name instead of the volume name to identify the volume. Enclose the wwn in angle brackets (< >).
cacheFlushModifier	Maximum amount of time that data for the volume remains in cache before being flushed to physical storage. Valid values are listed in the following “Notes” section.
cacheWithoutBatteryEnabled	Turns on or off caching without batteries. To turn caching without batteries on, set this parameter to TRUE . To turn caching without batteries off, set this parameter to FALSE .
mediaScanEnabled	Turns on or off media scan for the volume. To turn media scan on, set this parameter to TRUE . To turn media scan off, set this parameter to FALSE . (If media scan is disabled at the storage array level, this parameter has no effect.)
mirrorCacheEnabled	Turns on or off the mirror cache. To turn the mirror cache on, set this parameter to TRUE . To turn the mirror cache off, set this parameter to FALSE .
modificationPriority	Defines the priority for volume modifications while the storage array is operational. Valid entries are highest, high, medium, low, or lowest
owner	The controller that owns the volume. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Use this parameter only if you want to change the volume owner.

Parameter	Description
readCacheEnabled	Turns on or off the read cache. To turn the read cache on, set this parameter to TRUE . To turn the read cache off, set this parameter to FALSE .
writeCacheEnabled	Turns on or off the write cache. To turn the write cache on, set this parameter to TRUE . To turn the write cache off, set this parameter to FALSE .
readAheadMultiplier	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.
addCapacity	Increases storage size (capacity) of the volume for which you are defining properties. Size is defined in units of bytes, KB, MB, GB, or TB. The default units are bytes.
addDrives	Adds new physical disks to the volume. Specify the tray ID and slot ID for each physical disk you assign to the volume. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]). Use this parameter with the addCapacity parameter if you need to specify additional physical disks to accommodate the new size.
redundancyCheckEnabled	Turns on or off redundancy checking during a media scan. To turn redundancy checking on, set this parameter to TRUE . To turn redundancy checking off, set this parameter to FALSE .
segmentSize	The amount of data (in kilobytes) the controller writes on a single physical disk in a volume before writing data on the next physical disk. Use integer values.
userLabel	The new name you want to give an existing volume. Enclose the new volume name in double quotes (" ").
accessVolume	Use this parameter when you want to define the logical unit number for the access volume. (The logical unit number is the only property you can set for the access volume.
logicalUnitNumber	Defines specific volume-to-LUN mappings to an individual host or assigns the host to a host group.

.....

Parameter	Description
host	The name of the host to which the volume is mapped. Enclose the host name in double quotes (“ ”).
hostGroup	The name of the host group to which the volume is mapped. Enclose the host group name in double quotes (“ ”). defaultGroup is the host group that contains the host to which the volume is mapped.

Notes

When you use this command, you can specify on or more of the parameters. You do not, however, need to use all of the parameters.

You can apply the following parameters to only one volume at a time:

- addCapacity
- segmentSize
- userLabel
- logicalUnit Number

Setting addCapacity, addDrives, or segmentSize starts a long running operation that you cannot stop. These long running operations are performed in the background and do not prevent you from running other commands. To display the progress of long running operations, use the show volume actionProgress command.

The access volume is the volume in a SAN environment used for in-band communication between the storage management software and the storage array controller. This volume uses a LUN address and consumes 20 MB of storage space that is not available for application data storage. An access volume is required only for in-band managed storage arrays. If you specify accessVolume, the only property you can set is the logicalUnitNumber.

Valid values for the cache flush modifier are:

Immediate	Data is flushed as soon as it is placed into the cache.
250	Data is flushed after 250 ms.
500	Data is flushed after 500 ms.
750	Data is flushed after 750 ms.
1	Data is flushed after 1 s.
1500	Data is flushed after 1500 ms.
2	Data is flushed after 2 s.
5	Data is flushed after 5 s.
10	Data is flushed after 10 s.
20	Data is flushed after 20 s.
60	Data is flushed after 60 s (1 min.).
120	Data is flushed after 120 s (2 min.).
300	Data is flushed after 300 s (5 min.).
1200	Data is flushed after 1200 s (20 min.).
3600	Data is flushed after 3600 s (1 hr.).
Infinite	Data in cache is not subject to any age or time constraints. The data will be flushed based on other criteria managed by the controller.

Write caching without batteries enables write caching to continue if the controller batteries are completely discharged, not fully charged, or not present. If you set this parameter to `True` without an uninterruptable power supply (UPS) or other back-up power source, you can lose data if the storage array power fails. This parameter has no effect if write caching is disabled.

Modification priority defines the amount of system resources used when modifying volume properties. If you select the highest priority level, the volume modification uses the most system resource, which will decrease performance for host data transfers.

Cache read-ahead enables the controller to copy additional data blocks into cache while the controller reads and copies host requested data blocks from disk into cache. This increases the chance that a future request for data can be fulfilled from cache. Cache read-ahead is important for multimedia applications that use sequential data transfers. Choosing

.....

a higher cache read-ahead multiplier might increase the cache hit percentage, however, transfer time might also increase.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfers.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Minimum Firmware Level

`addCapacity` parameter 5.00

Set Volume Copy

This command defines the properties for a volume copy pair.

Syntax

```
set volumeCopy target [targetName] [source [sourceName]]
copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)
```

Parameters

Parameter	Description
target	The name of the target volume for which you want to define properties. Enclose the target name in square brackets ([]). If the target name has more than one word, white spaces, or special characters, you must also enclose the target volume name in double quotes (“ ”).
source	The name of the source volume for which you want to define properties. Enclose the source volume name in square brackets ([]). If the source volume name has more than one word, white spaces, or special characters, you must also enclose the source volume name in double quotes (“ ”).
copyPriority	Defines the priority volume copy will have relative to host I/O activity. Valid entries are highest , high , medium , low , or lowest
targetReadOnlyEnabled	Defines whether you can write to the target volume or only read from the target volume. To be able to write to the target volume, set this parameter to FALSE . To prevent writing to the target volume, set this parameter to TRUE .

Notes

When you use this command, you can specify on or more of the parameters. You do not, however, need to use all of the parameters.

Minimum Firmware Level

5.40

Set Volume Group

This command defines the properties for a volume group.

Syntax

```
set volumeGroup [volumeGroupName]
```

.....

```
addDrives=(trayID1,slotID1 ... trayIDn,slotIDn) raidLevel=(0
| 1 | 3 | 5) owner=(a | b) availability=(online | offline)
```

Parameters

Parameter	Description
volumeGroupNumber	The sequence number of the volume group for which you want to set properties. Enclose the volume group number in square brackets ([]).
trayID,slotID	Identifies the physical disk by module and slot location that you want to include in the volume group. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in parentheses.
raidLevel	The RAID level for the volume group. Valid values are 0, 1, 3, or 5 .
owner	The controller that owns the volume group. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Use this parameter only if you want to change the volume group owner.
availability	Sets the volume group online or offline.

Notes

When you use this command, you can specify on or more of the parameters. You do not, however, need to use all of the parameters.

Setting `addDrives` or `raidLevel` starts a long operation that cannot be stopped.

Minimum Firmware Level

6.10

Show Controller

For each controller in a storage array, this command returns:

- Status (online, offline)
- Current firmware and NVSRAM configuration

-
- Pending firmware and NVSRAM configuration configurations (if any)
 - Board ID
 - Product ID
 - Product revision
 - Serial Number
 - Date of manufacture
 - Cache/processor size
 - Date/time to which the controller is set
 - Associated volumes (including preferred owner)
 - Ethernet port
 - physical disk interface
 - Host interface (this applies only to Fibre Channel host interfaces)

Syntax

```
show (allControllers | controller [(a | b)]) [summary]
```

Parameters

Parameter	Description
<code>allControllers</code>	Returns information about both controllers in the storage array.
<code>(a b)</code>	Returns information about a specific controller in the storage array. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]).
<code>summary</code>	Returns a concise list of information about both controllers in the storage array.

Notes

The following list is an example of the information returned by the `show controller` command. This example is used only to show how the information is presented and should not be considered to represent best practice for a storage array configuration.

```
.....

Controller in Slot A
Status: Online
Current configuration
  Firmware version: 96.10.21.00
  Appware version: 96.10.21.00
  Bootware version: 96.10.21.00
  NVSRAM version: N4884-610800-001
Pending configuration
  Firmware version: Not applicable
  Appware version: Not applicable
  Bootware version: Not applicable
  NVSRAM version: Not applicable
  Transferred on: Not applicable
Board ID: 4884
Product ID: INF-01-00
Product revision: 9610
Serial number: 1T14148766
Date of manufacture: October 14, 2001
Cache/processor size (MB): 1024/128
Date/Time: Wed Feb 18 13:55:53 MST 2004
Associated Volumes (* = Preferred Owner):
  1*, 2*, CTL 0 Mirror Repository*, Mirror Repository 1*,
  JCG_Remote_MirrorMenuTests*
Ethernet port: 1
  MAC address: 00:a0:b8:0c:c3:f5
  Host name: ausctlr9
  Network configuration: Static
  IP address: 172.22.4.249
  Subnet mask: 255.255.255.0
  Gateway: 172.22.4.1
  Remote login: Enabled
Drive interface: Fibre
  Channel: 1
  Current ID: 125/0x1
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
Drive interface: Fibre
  Channel: 2
  Current ID: 125/0x1
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
```

```

.....

Drive interface: Fibre
  Channel: 3
  Current ID: 125/0x1
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
Drive interface: Fibre
  Channel: 4
  Current ID: 125/0x1
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
Host interface: Fibre
  Port: 1
  Current ID: Not applicable/0xFFFFFFFF
  Preferred ID: 126/0x0
  NL-Port ID: 0x011100
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
  Topology: Fabric Attach
  World-wide port name: 20:2c:00:a0:b8:0c:c3:f6
  World-wide node name: 20:2c:00:a0:b8:0c:c3:f5
  Part type: HPFC-5200   revision 10
Host interface: Fibre
  Port: 2
  Current ID: Not applicable/0xFFFFFFFF
  Preferred ID: 126/0x0
  NL-Port ID: 0x011000
  Maximum data rate: 2 Gbps
  Current data rate: 1 Gbps
  Data rate control: Switch
  Link status: Up
  Topology: Fabric Attach
  World-wide port name: 20:2c:00:a0:b8:0c:c3:f7
  World-wide node name: 20:2c:00:a0:b8:0c:c3:f5
  Part type: HPFC-5200   revision 10

```

When you use the `summary` parameter, the command returns the list of information without the physical disk channel and host channel information.

The detailed controller information is also returned for the `show storageArray` command.

Minimum Firmware Level

`summary` parameter 5.43

Show Controller NVSRAM

This command returns the NVSRAM bytes for the specified host type. If you do not enter the optional parameters, this command returns the entire NVSRAM. Refer to “Show Controller NVSRAM” on page A-17 to see an example of a table of NVSRAM values returned by this command.

Syntax

```
show (allControllers | controller [(a | b)]) NVSRAM  
[hostType=(hostTypeIndexLabel | host="hostName")]
```

Parameters

Parameter	Description
allControllers	Returns information about both controllers in the storage array.
(a b)	Returns information about a specific controller in the storage array. Valid controller identifiers are a or b where a is the controller in slot A and b is the controller in slot B. Enclose the controller identifier in square brackets ([]).
hostTypeIndexLabel	The index label or number of the host type. Use the <code>show storageArray hostTypeTable</code> command to generate a list of available host type identifiers.
hostName	The name of the host connected to the controllers. Enclose the host name in double quotes (“”).

Notes

Use the `show controller NVSRAM` command to display parts or all of the NVSRAM before using the `set controller` command to change NVSRAM values.

Minimum Firmware Level

6.10

Show Drive

For each physical disk in the storage array, this command returns:

- Total number of physical disks
- Type of physical disk (Fibre, SATA, PATA)
- Basic physical disk information:
 - Module and slot location
 - Status
 - Capacity
 - Data transfer rate
 - Product ID
 - Firmware level
- Physical disk channel information:
 - Module and slot location
 - Preferred channel
 - Redundant channel
- Hot spare coverage
- Details for each physical disk

Depending on the size of your storage array, this information can be several pages long. Refer to “Show Drive” on page A-20 to see an example of the physical disk information returned by `show drives` command. The physical disk information is also returned for the `show storageArray profile` command.

Syntax

```
show (allDrives [driveType=(fibre | SATA | PATA)] | drive
[trayID,slotID] | drives [trayID1,slotID1 ...
trayIDn,slotIDn]) [summary]
```

Parameters

Parameter	Description
<code>allDrives</code>	Returns information about all physical disks in the storage array.
<code>driveType</code>	The type of physical disks for which you want to retrieve information. Valid physical disk types are: Fibre, SATA, or PATA.
<code>drives</code>	Identifies the module and slot where the physical disk resides. You can enter tray IDs and slot IDs for one or several physical disks. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in parentheses.
<code>summary</code>	Returns the status, capacity, data rate, product ID, and firmware version for the specified drives.

Notes

To determine information about the type and location of all physical disks in the storage array, use the `allDrives` parameter.

To determine the information about the Fibre, SATA, or PATA physical disks in the storage array use the `driveType` parameter.

To determine the type of physical disk in a specific location, use the `drive` parameter and enter the tray and slot IDs for the physical disk.

Minimum Firmware Level

5.43

Show Drive Channel Stats

This command displays cumulative physical disk channel data transfer and error information. If the controller has automatically degraded a channel, this command also displays interval statistics. When using this command, you can display information about one specific physical disk channel, several physical disk channels, or all physical disk channels. To see an example showing a table returned by this command, refer to “Show Drive Channel Status” on page A-26.

Syntax

```
show (driveChannel [(1 | 2 | 3 | 4)] | driveChannels [(1 | 2 | 3 | 4) ... (1n | 2n | 3n | 4n)] | allDriveChannels) stats
```

Parameters

Parameter	Description
driveChannel	The identifier number of the physical disk channel for which you want to display information. Valid physical disk channel values are 1 , 2 , 3 , or 4 . Enclose the physical disk channels in square brackets ([]).

Minimum Firmware Level

6.10

Show Drive Download Progress

This command returns the status of firmware downloads for the physical disks targeted by the `download drive firmware` or `download storageArray driveFirmware` commands.

Syntax

```
show allDrives downloadProgress
```

Parameters

None.

Notes

When all firmware downloads have successfully completed, this command returns good status. If any firmware downloads fail, this command displays the firmware download status of each physical disk that was targeted. This command returns the following status:

Successful	The downloads completed without errors.
Not Attempted	The downloads did not start.
Partial Download	The download is in progress.
Failed	The downloads completed with errors.

Minimum Firmware Level

6.10

Show Host Ports

For all host ports connected to a storage array, this command returns the following information:

- Host port identifier
- Host port name
- Host type

Syntax

```
show allHostPorts
```

Parameters

None.

Notes

This command returns host port information similar to the following example:

```
HOST PORT IDENTIFIER    HOST PORT NAME    HOST TYPE
12:34:56:54:33:22:22:22  Jupiter1          Solaris
12:34:56:78:98:98:88:88  Pluto1           Windows 2000/Server 2003 Clustered
54:32:12:34:34:55:65:66  Undefined         Undefined
```

Minimum Firmware Level

5.40

Show Remote Volume Mirror Volume Candidates

This command returns information about the candidate volumes on the remote storage array that you can use as secondary volumes for a primary volume.

Syntax

```
show remoteMirror candidates primary="volumeName"  
remoteStorageArrayName= "storageArrayName"
```

Parameters

Parameter	Description
primary	Name of the local volume that you want for the primary volume in the Remote Volume Mirror pair. Enclose the volume name in double quotes (" ").
remoteStorageArrayName	The remote storage array that contains possible volumes for a secondary volume. If the storage array name has more than one word, white spaces, or special characters, you must also enclose the host name in double quotes (" ").

Show Remote Volume Mirror Volume Synchronization Progress

This command returns the progress of data synchronization between the primary volume and secondary volume in a Remote Volume Mirror. This command displays the progress as a percentage of data synchronization that has been completed.

Syntax

```
show remoteMirror (localVolume ["volumeName"] | localVolumes  
["volumeName1" ... "volumeNameN"]) synchronizationProgress
```

.....

Parameters

Parameter	Description
localVolume	Name of the primary volume of the Remote Volume Mirror pair for which you want to check synchronization progress. Enclose the primary volume name in double quotes (“ ”) inside square brackets ([]).

Show Storage Array

This command returns configuration information about the storage array. The parameters return lists of values for the components and features in the storage array. You can enter the command with a single parameters or more than one parameter. If you enter the command without any parameters, the entire storage array profile is displayed (which is the same information as if you entered the `profile` parameter).

Syntax

```
show storageArray profile batteryAge defaultHostType  
healthStatus hostTypeTable hotSpareCoverage features time  
volumeDistribution [summary]
```

Parameters

Parameter	Description
profile	Displays all properties of the logical and physical components that comprise the storage array. The information returned takes several screens to display.
batteryAge	Displays the, status, age of the battery in days, and the number of days until the battery needs to be replaced.
defaultHostType	Displays the default host type and host type index.
healthStatus	Displays the health, logical properties, and physical component properties of the storage array.
hostTypeTable	Displays a table of all host types known to the controller. Each row in the table displays a host type index and the platform the index represents.

Parameter	Description
hotSpareCoverage	Displays information about which volumes of the storage array have hot spare coverage and which volumes do not.
features	Displays a list of the feature identifiers for all enabled features in the storage array.
time	Displays the current time to which both controllers in the storage array are set.
volumeDistribution	Displays the current controller owner for each volume in the storage array.

Notes

The `profile` parameter returns detailed information about the storage array. The information covers several screens on a display. You may need to increase the size of your display buffer to see all the information. Because this information is so detailed, you may want to save the output of this parameter to a file. To save the output to a file enter the `show storageArray` command similar to the following:

```
c:\...\sm9\client>smcli 123.45.67.89 -c "show storagearray
profile;" -o "c:\\folder\\storagearray profile.txt"
```

(The previous command syntax is for a host running a Microsoft operating system. The actual syntax will vary depending on your operating system.)

“Show Storage Array” on page A-2 shows the type of information returned. When you save the information to a file you can use the information as a record of your configuration and as an aid during recovery.

The `batteryAge` parameter returns information in the form:

```
Battery status: Optimal
Age: 1 day(s)
Days until replacement: 718 day(s)
```

The `defaultHostType` parameter returns information in the form:

```
Default host type: Linux (Host type index 6)
```

.....

The `healthStatus` parameter returns information in the form:

```
storage array health status = optimal.
```

The `hostTypeTable` parameter returns information in the form:

NVSRAM HOST TYPE INDEX DEFINITIONS

INDEX	AVT STATUS	TYPE
0	Disabled	Windows NT Non-Clustered (SP5 or higher)
1 (Default)	Disabled	Windows 2000/Server 2003 Non-Clustered
2	Disabled	Solaris
3	Enabled	HP-UX
4	Disabled	AIX
5	Disabled	Irix
6	Enabled	Linux
7	Disabled	Windows NT Clustered (SP5 or higher)
8	Disabled	Windows 2000/Server 2003 Clustered
9	Enabled	Netware Non-Failover
10	Enabled	PTX
11	Enabled	Netware Failover
12	Enabled	Solaris (with Veritas DMP)

The `hotSpareCoverage` parameter returns information in the form:

```
The following volume groups are not protected: 2, 1
Total hot spare drives: 0
  Standby: 0
  In use: 0
```

The `features` parameter returns information in the form:

```
storagePartitionMax
snapshot
remoteMirror
volumeCopy
```

The `time` parameter returns information in the form:

```
Controller in Slot A
Date/Time: Thu Jun 03 14:54:55 MDT 2004
Controller in Slot B
Date/Time: Thu Jun 03 14:54:59 MDT 2004
```

.....

The `volumeDistribution` parameter returns information in the form:

```
Volume name: 10 Current owner is controller in slot: A
Volume name: CTL 0 Mirror Repository0 Current owner is controller in slot: A
Volume name: Mirror Repository 10 Current owner is controller in slot: A
Volume name: 20 Current owner is controller in slot: A
Volume name: JCG_Remote_MirrorMenuTests0 Current owner is controller in slot: A
```

Minimum Firmware Level

`defaultHostType` parameter 5.00

`summary` parameter 5.43

Show Storage Array Auto Configure

This command displays the default auto configuration that the storage array will create if you issue the `autoConfigure storageArray` command. If you want to determine whether the storage array can support specific properties, enter the parameter for the properties when you issue this command. You do not, however, need to enter any parameters for this command to return configuration information. If you do not specify any properties, this command returns the RAID 5 candidates for each physical disk type. If RAID 5 candidates are not available, this command returns candidates for RAID 3, RAID 1, or RAID 0. When you specify auto configuration properties, the controllers will validate that the firmware will support the properties.

Syntax

```
show storageArray autoConfiguration [driveType=(fibre | SATA
| PATA) raidLevel=(0 | 1 | 3 | 5)
volumeGroupWidth=numberOfDrives
volumeGroupCount=numberOfVolumeGroups
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotspares segmentSize=segmentSizeValue
readAheadMultiplier=multiplierValue]
```

.....

Parameters

Parameter	Description
driveType	The type of physical disks that you want to use for the storage array. Valid physical disk types are: Fibre , SATA , or PATA . The <code>driveType</code> parameter is not required if only one type of physical disk is in the storage array.
raidLevel	The RAID level of the volume group that contains the physical disks in the storage array. Valid RAID levels are 0 , 1 , 3 , or 5 .
volumeGroupWidth	The number of physical disks in a volume group in the storage array. The number of physical disks in a volume group depends on the capacity of the physical disks. Integer values are required.
volumeGroupCount	The number of volume groups in the storage array. Use integer values.
volumesPerGroupCount	The number of equal capacity volumes per volume group. Use integer values.
hotSpareCount	The number of hot spares you want in the storage array. Use integer values.
segmentSize	The amount of data (in kilobytes) the controller writes on a single drive in a volume before writing data on the next drive. Use integer values.
readAheadMultiplier	Defines how many additional data blocks are read into cache. Valid values range from 0 to 65535.

Notes

A volume group is a set of physical disks logically grouped together by the controllers in the storage array. The number of physical disks in a volume group depends on the size of physical disks available for the volume group. The following list relates physical disk capacity to the number of physical disks in a volume group.

- 18 and 36 GB physical disks, 30 physical disks maximum in the volume group
- 73 GB physical disks, 28 physical disks maximum in the volume group
- 180 GB physical disks, 11 physical disks maximum in the volume group

.....

A 73 or 180 GB physical disk will not report exactly 73 or 180 GB as its capacity. The limitation for the number of large physical disks in a volume group is the two terabyte size restriction.

Hot spare physical disks can replace any failed physical disk in the storage array. A hot spare physical disk must have capacity greater than or equal to any physical disk that can fail. If a hot spare physical disk is smaller than a failed physical disk, the hot spare physical disk cannot be used to rebuild the data from the failed physical disk. Hot spare physical disks are available only for RAID level 1, 3, or 5. The maximum number of SCSI hot spare physical disks is equal to the number of SCSI physical disk channels supported by the controller. The maximum number of Fibre Channel hot spare physical disks per storage array is 16.

The size of a segment determines how many data blocks the controller writes on a single physical disk in a volume before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many blocks it contains. For example, an 8 KB segment holds 16 data blocks; a 64 KB segment holds 128 data blocks. When you enter a value for the segment size, the value is checked against the supported values provided by the controller at run time. If the value you enter is not valid, the controller returns a list of valid values. For optimal performance in a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy a data transfer request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests. If the volume is in a single-user large data transfer environment (such as multi-media) performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size multiplied by the number of physical disks in the volume group that are used for data transfers.) In this case, multiple disks are used for the same request, but each disk is only accessed once.

Minimum Firmware Level

6.10

Show Storage Array Host Topology

This command returns storage partition topology, host type labels, and host type index for the host storage array.

Syntax

```
show storageArray hostTopology
```

Parameters

None.

Notes

This command returns host topology information similar to the following example:

```
TOPOLOGY DEFINITIONS
DEFAULT GROUP
Default type: Windows 2000/Server 2003 Non-Clustered
Host Group: scott
Host: scott1
Host Port: 28:37:48:55:55:55:55
Alias: scott11
Type: Windows 2000/Server 2003 Clustered
Host: scott2
Host Port: 98:77:66:55:44:33:21:23
Alias: scott21
Type: Windows 2000/Server 2003 Clustered
Host: Bill
Host Port: 12:34:55:67:89:88:88:88
Alias: Bill1
Type: Windows 2000/Server 2003 Non-Clustered

NVSRAM HOST TYPE INDEX DEFINITIONS

INDEX    AVT STATUS  TYPE
0        Disabled  Windows NT Non-Clustered (SP5 or higher)
1 (Default) Disabled  Windows 2000/Server 2003 Non-Clustered
2        Disabled  Solaris
3        Enabled   HP-LUX
4        Disabled  AIX
5        Disabled  Irix
6        Enabled   Linux
7        Disabled  Windows NT Clustered (SP5 or higher)
8        Disabled  Windows 2000/Server 2003 Clustered
9        Enabled   Netware Non-Failover
10       Enabled   PTX
11       Enabled   Netware Failover
12       Enabled   Solaris (with Veritas DMP)
```

Minimum Firmware Level

5.20

Show Storage Array LUN Mappings

This command returns information from the storage array profile about the storage array LUN mappings. Default group LUN mappings are always displayed. If you run this command without any parameters, this command returns all LUN mappings.

Syntax

```
show storageArray lunMappings [host ["hostName"] | hostgroup
["hostGroupName"]]
```

.....

Parameters

Parameter	Description
host	The name of a specific host for which you want to see the LUN mappings. Enclose the host name in double quotes (“ ”) inside square brackets ([]).
hostGroup	The name of a specific host group for which you want to see the LUN mappings. Enclose the host group name in double quotes (“ ”) inside square brackets ([]).

Notes

This command returns host topology information similar to the following example:

```
MAPPINGS (SANshare Storage Partitioning - Enabled (0 of 16 used))-----
```

```
VOLUME NAME LUN CONTROLLER ACCESSIBLE BY VOLUME STATUS
Access Volume 7 A,B Default Group Optimal
21 21 B Default Group Optimal
22 22 B Default Group Optimal
23 23 B Default Group Optimal
24 24 B Default Group Optimal
25 25 B Default Group Optimal
26 26 B Default Group Optimal
27 27 B Default Group Optimal
28 28 B Default Group Optimal
29 29 B Default Group Optimal
30 30 B Default Group Optimal
31 31 B Default Group Optimal
32 32 B Default Group Optimal
33 33 B Default Group Optimal
34 34 B Default Group Optimal
35 35 B Default Group Optimal
36 36 B Default Group Optimal
37 37 B Default Group Optimal
38 36 B Default Group Optimal
39 39 B Default Group Optimal
40 40 B Default Group Optimal
```

Minimum Firmware Level

6.10

Show Storage Array Unreadable Sectors

This command returns a table of the addresses of all sectors in the storage array that cannot be read. The table is organized with column headings for the following information.

1. Volume user label
2. LUN
3. Accessible By
 - Host or host group
4. Date/Time
5. Volume-relative logical block address
 - Hex format (0xxxxxxxxx)
6. physical disk Location
 - Tray t, Slot s
7. physical disk-relative logical block address
 - Hex format (0xxxxxxxxx)
8. Failure Type

The data is sorted first by volume user label and second by LBA. Each entry in the table corresponds to a single sector.

Syntax

```
show storageArray unreadableSectors
```

Parameters

None.

Minimum Firmware Level

6.10

Show String

This command shows a string of text from a script file. This command is similar to the echo command in MS DOS and UNIX.

Syntax

```
show "string"
```

Parameters

None.

Notes

Enclose the string in double quotes (“ ”).

Minimum Firmware Level

6.10

Show Volume

For the volumes in a storage array, this command returns:

- Number of volumes
- Name
- Status
- Capacity
- RAID level
- Volume group where the volume is located
- Details
 - Volume ID
 - Subsystem ID
 - physical disk type (Fibre, SATA, PATA)
 - Tray loss protection
 - Preferred owner
 - Current owner

-
- Segment size
 - Modification priority
 - Read cache status (enabled, disabled)
 - Write cache status (enabled, disabled)
 - Write cache without batteries status (enabled, disabled)
 - Write cache with mirroring status (enabled, disabled)
 - Flush write cache after time
 - Cache read ahead multiplier
 - Enable background media scan status (enabled, disabled)
 - Media scan with redundancy check status (enabled, disabled)
- Snapshot repository volumes
 - Mirror repository volumes
 - Snapshot volumes
 - Copies

To see an example of the information returned by this command, refer to “Show Volumes” on page A-27.

Syntax

```
show (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) [summary]
```

Parameters

Parameter	Description
<code>allVolumes</code>	Use this parameter to return information about all volumes in the storage array.
<code>volumeName</code>	Name of the specific volume for which you are retrieving information. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).
<code>summary</code>	Returns a concise list of information about the volumes.

Minimum Firmware Level

5.00

`summary` parameter 5.43

Show Volume Action Progress

For a long-running operation that is currently running on a volume, this command returns information about the volume action and amount of the long-running operation completed. The amount of the long-running operation that is completed is shown as a percentage (for example, 25 means that 25 percent of the long-running operation is completed).

Syntax

```
show volume ["volumeName"] actionProgress
```

Parameters

Parameter	Description
<code>volumeName</code>	Name of the volume running the long running operation. Enclose the volume name in double quotes (“ ”) inside square brackets ([]).

Minimum Firmware Level

5.43

Show Volume Copy

This command returns information about volume copy operations. The information returned is

- Copy status
- Start timestamp
- Completion timestamp
- Copy priority
- Source or target volume WWN
- Target volume read-only attribute setting
- Percent completed, if a volume copy operation is in progress

You can retrieve information about a specific volume copy pair or all volume copy pairs in the storage array.

Syntax

```
show volumeCopy (allVolumes | source ["sourceName"] | target  
["targetName"])
```

Parameters

Parameter	Description
allVolumes	Use this parameter to return information about volume copy operations for all volume copy pairs.
source	The name of the source volume about which you want to retrieve information. Enclose the source volume name in double quotes (“”) inside square brackets ([]).
target	The name of the target volume about which you want to retrieve information. Enclose the target volume name in double quotes (“”) inside square brackets ([]).

Minimum Firmware Level

5.40

Show Volume Copy Source Candidates

This command returns information about the candidate volumes that you can use as the source for a volume copy operation.

Syntax

```
show volumeCopy sourceCandidates
```

Parameters

None.

Notes

This command returns volume copy source information shown in the following example:

```
Volume Name: 1
  Capacity(GB): 1.000
  Volume Group: 1
Volume Name: RAID1testusingGUI
  Capacity(GB): 1.000
  Volume Group: 2
Volume Name: Test
  Capacity(GB): 1.000
  Volume Group: 3
Volume Name: 2
  Capacity(GB): 1.000
  Volume Group: 3
Volume Name: 1S1
  Capacity(GB): 1.000
  Volume Group: none
```

Minimum Firmware Level

6.10

Show Volume Copy Target Candidates

This command returns information about the candidate volumes that you can use as the target for a volume copy operation.

Syntax

```
show volumeCopy source ["sourceName"] targetCandidates
```

Parameters

Parameter	Description
source	The name of the source volume for which you are trying to find a candidate target volume. Enclose the source volume name in double quotes (“ ”) inside square brackets ([]).

Minimum Firmware Level

6.10

Show Volume Group

This command returns the following information about a volume group:

- Status (online or offline)
- Drive type (Fibre, SATA, or PATA)
- Tray loss protection (yes or no)
- Current owner (controller slot A or slot B)
- Associated volumes and free capacity
- Associated physical disks (drives)

NOTE You can use the free capacity area value when creating a volume based on the free capacity of a volume group. For a description of how to use the free capacity value, refer to the `create volume` command on page 8-20.

Syntax

show volumeGroup [*volumeGroupName*]

Parameters

Parameter	Description
volumeGroupName	The number of the volume group for which you want to display information. Enclose the volume group number in square brackets ([]).

Notes

This command returns volume group information as shown in the following example:

```
Volume Group 1 (RAID 5)
  Status: Online
  Drive type: Fibre Channel
  Tray loss protection: No
  Current owner: Controller in slot A
  Associated volumes and free capacities:
    1 (1 GB), 1R1 (0.2 GB), Free Capacity (134.533 GB)
  Associated drives (in piece order):
    Drive at Tray 1, Slot 14
    Drive at Tray 1, Slot 13
    Drive at Tray 1, Slot 12
```

Minimum Firmware Level

6.10

Show Volume Performance Statistics

This command returns information about the performance of the volumes in a storage array.

Syntax

```
show (allVolumes | volume [volumeName] | volumes
[volumeName1 ... volumeNameN] performanceStats
```

.....

Parameters

Parameter	Description
allVolumes	Use this parameter to return performance statistics about all volumes in the storage array.
volumeName	Name of the specific volume for which you are retrieving performance statistics. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Notes

Before issuing the `show volume performanceStat` command, issue the `set session performanceMonitorInterval` and `set session performanceMonitorIterations` commands to define how often you collect the statistics.

This command returns volume performance statistics as shown in the following example:

```
Performance Monitor Statistics for Storage Array: ausctrl9
Date/Time: 2/19/04 4:03:09 PM
Polling interval in seconds: 10

Devices,Total,Read,Cache Hit,Current,Maximum,Current,Maximum
,IOs,Percentage,Percentage,KB/second,KB/second,IO/second,IO/second
Capture Iteration: 1
Date/Time: 2/19/04 4:03:09 PM
CONTROLLER IN SLOT A,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
Volume 1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
Volume 2,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
Volume 3,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
Volume JCG_Remote_MirrorMenuTests,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
STORAGE ARRAY TOTALS,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
```

Minimum Firmware Level

6.10

Show Volume Reservations

This command returns information about the volumes that have reservations.

Syntax

```
show (allVolumes | volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) reservations
```

Parameters

Parameter	Description
allVolumes	Use this parameter to return reservation information about all volumes in the storage array.
volumeName	Name of the specific volume for which you are retrieving reservation information. You can enter more than one volume name. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (" ").

Minimum Firmware Level

5.40

Start Drive Channel Locate

This command identifies the physical disk modules connected to a specific physical disk channel by turning on the indicator lights for the physical disk module connected. (Use the `stop driveChannel locate` command to turn off the physical disk module indicator lights.)

Syntax

```
start driveChannel [(1 | 2 | 3 | 4)]locate
```

.....

Parameters

Parameter	Description
driveChannel	The identifier number of the physical disk channel that you want to locate. Valid physical disk channel values are 1, 2, 3, or 4 . Enclose the physical disk channel identifier number in square brackets ([]).

Minimum Firmware Level

6.10

Start Drive Initialize

This command starts physical disk initialization.

CAUTION Potential storage array configuration damage. This command will destroy user data.

Syntax

```
start drive [trayID,slotID] initialize
```

Parameters

Parameter	Description
trayID,slotID	Identifies the module and slot where the physical disk resides. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).

Minimum Firmware Level

6.10

Start Drive Locate

This command locates a physical disk by turning on the physical disk indicator lights.
(Use the `stop drive locate` command to turn off the physical disk indicator light.)

Syntax

```
start drive [trayID,slotID] locate
```

Parameters

Parameter	Description
<code>trayID,slotID</code>	Identifies the module and slot where the physical disk resides. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).

Minimum Firmware Level

6.10

Start Drive Reconstruction

This command starts reconstructing a physical disk.

Syntax

```
start drive [trayID,slotID] reconstruct
```

Parameters

Parameter	Description
<code>trayID,slotID</code>	Identifies the module and slot where the physical disk reside. Tray ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the tray ID and slot ID values in square brackets ([]).

Minimum Firmware Level

5.43

Start Remote Volume Mirror Synchronization

This command starts Remote Volume Mirror synchronization.

Syntax

```
start remoteMirror primary ["volumeName"] synchronize
```

Parameters

Parameter	Description
volumeName	Name of the primary volume for which you want to start synchronization. Enclose the primary volume name in double quotes (“ ”) inside square brackets ([]).

Minimum Firmware Level

6.10

Start Storage Array Locate

This command locates a storage array by turning on the indicator lights for the storage array. (Use the `stop storageArray locate` command to turn off the indicator lights for the storage array.)

Syntax

```
start storageArray locate
```

Parameters

None.

Minimum Firmware Level

6.10

Start Tray Locate

This command locates a module by turning on the indicator lights. (Use the `stop tray locate` command to turn off the indicator lights for the module.)

Syntax

```
start tray [trayID] locate
```

Parameters

Parameter	Description
<code>trayID</code>	Identifies the module you want to locate. Tray ID values are 0 to 99. Enclose the tray ID value in square brackets ([]).

Minimum Firmware Level

6.10

Start Volume Group Defragment

This command starts a defragment operation on the specified volume group.

NOTE Defragmenting a volume group starts a long running operation that you cannot stop.

Syntax

```
start volumeGroup [volumeGroupNumber] defragment
```

.....

Parameters

Parameter	Description
volumeGroupName	The number of the volume group that you want to defragment. Enclose the volume group number in square brackets ([]).

Minimum Firmware Level

6.10

Start Volume Initialization

This command starts the formatting of a volume in a storage array.

NOTE Initializing a volume is a long-running operation that you cannot stop.

Syntax

```
start volume [volumeName] initialize
```

Parameters

Parameter	Description
volumeName	Name of the volume for which you are starting the formatting. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (“ ”).

Minimum Firmware Level

6.10

Stop Drive Channel Locate

This command turns off the physical disk module indicator lights that were turned on by the `start driveChannel locate` command.

Syntax

```
stop driveChannel locate
```

Parameters

None.

Minimum Firmware Level

6.10

Stop Drive Locate

This command turns off the physical disk indicator lights that were turned on by the `start drive locate` command.

Syntax

```
stop drive locate
```

Parameters

None.

Minimum Firmware Level

6.10

Stop Snapshot

This command stops a copy-on-write operation.

Syntax

```
stop snapshot (volume [volumeName] | volumes [volumeName1  
... volumeNameN])
```

Parameters

Parameter	Description
volumeName	The name of the specific volume for which you want to stop a copy-on-write operation. You can enter more than one volume name. Enclose the volume names in square brackets ([]). If the volume names have more than one word, white spaces, or special characters, you must also enclose the volume names in double quotes (“ ”).

Minimum Firmware Level

6.10

Stop Storage Array Drive Firmware Download

This command stops a firmware download to the physical disks in a storage array that was started with the `download storageArray driveFirmware` command. This command does not stop a firmware download that is already in progress to a physical disk; however, this command stops all firmware downloads to physical disks that are waiting for the download.

Syntax

```
stop storageArray driveFirmwareDownload
```

Parameters

None.

Minimum Firmware Level

6.10

Stop Storage Array Locate

This command turns off the storage array indicator lights that were turned on by the `start storageArray locate` command.

Syntax

```
stop storageArray locate
```

Parameters

None.

Minimum Firmware Level

6.10

Stop Tray Locate

This command turns off the module indicator lights that were turned on by the `start tray locate` command.

Syntax

```
stop tray locate
```

Parameters

None.

Minimum Firmware Level

6.10

Stop Volume Copy

This command stops a volume copy operation.

Syntax

```
stop volumeCopy target [targetName] [source [sourceName]]
```

Parameters

Parameter	Description
target	The name of the target volume for which you want to stop a volume copy operation. Enclose the target volume name in square brackets ([]). If the target volume name has more than one word, white spaces, or special characters, you must also enclose the target volume name in double quotes (“ ”).
source	The name of the source volume for which you want to stop a volume copy operation. Enclose the source volume name in square brackets ([]). If the source volume name has more than one word, white spaces, or special characters, you must also enclose the source volume name in double quotes (“ ”).

Minimum Firmware Level

5.40

Suspend Remote Volume Mirror

This command suspends a Remote Volume Mirror operation.

Syntax

```
suspend remoteMirror (volume [volumeName] | volumes  
[volumeName1 ... volumeNameN]) writeConsistency=(TRUE |  
FALSE)
```

Parameters

Parameter	Description
volumeName	Name of the volume for which you want to suspend operation. Enclose the volume name in square brackets ([]). If the volume name has more than one word, white spaces, or special characters, you must also enclose the volume name in double quotes (" ").
writeConsistency	Defines whether the volumes identified in this command are in a write-consistency group or are separate. For the volumes in the same write-consistency group, set this parameter to TRUE . For the volumes that are separate, set this parameter to FALSE .

Notes

If you set `writeConsistency` to `TRUE`, the volumes must be in a write-consistency group (or groups). The action of the command is to suspend all consistency groups containing the volumes. For example, if volumes A, B, and C are in a consistency group and they have remote counterparts A', B' and C', the command `suspend remoteMirror volue["A"] writeConsistency=true` suspends A-A', B-B', and C-C'. If you have a consistency group 1={A, B, C} and consistency group 2={D, E, F}, the command `suspend remoteMirror volumes["A", "D"] writeConsistnecy=true` suspends both consistency groups.

Commands Listed by Function

Controller Commands

- “Clear Drive Channel Statistics” on page 8-12
- “Diagnose Controller” on page 8-37
- “Enable Controller” on page 8-44
- “Reset Controller” on page 8-56
- “Save Controller NVSRAM” on page 8-60
- “Set Controller” on page 8-66
- “Set Drive Channel Status” on page 8-68
- “Set Host Channel” on page 8-72
- “Show Controller” on page 8-89
- “Show Controller NVSRAM” on page 8-93
- “Show Drive Channel Stats” on page 8-95
- “Start Drive Channel Locate” on page 8-116
- “Stop Drive Channel Locate” on page 8-122

Host Topology Commands

- “Create Host” on page 8-15
- “Create Host Group” on page 8-16
- “Create Host Port” on page 8-17
- “Delete Host” on page 8-33
- “Delete Host Group” on page 8-34
- “Delete Host Port” on page 8-34
- “Set Host” on page 8-70
- “Set Host Group” on page 8-72
- “Set Host Port” on page 8-73
- “Show Host Ports” on page 8-97

Physical Disk (Drive) Commands

- “Download Drive Firmware” on page 8-39
- “Revive Drive” on page 8-59
- “Save Drive Log” on page 8-61
- “Set Drive Hot Spare” on page 8-69
- “Set Drive State” on page 8-69
- “Show Drive” on page 8-94
- “Show Drive Download Progress” on page 8-96
- “Start Drive Initialize” on page 8-117
- “Start Drive Locate” on page 8-118
- “Start Drive Reconstruction” on page 8-118
- “Stop Drive Locate” on page 8-122

Remote Volume Mirror Commands

- “Activate Remote Volume Mirroring Feature” on page 8-5
- “Create Remote Volume Mirror” on page 8-26
- “Deactivate Remote Volume Mirror” on page 8-32
- “Diagnose Remote Mirror” on page 8-38
- “Re-create Remote Volume Mirroring Repository” on page 8-49
- “Remove Remote Volume Mirror” on page 8-52
- “Resume Remote Volume Mirror” on page 8-58
- “Set Remote Volume Mirror” on page 8-74
- “Show Remote Volume Mirror Volume Candidates” on page 8-98
- “Show Remote Volume Mirror Volume Synchronization Progress” on page 8-98
- “Start Remote Volume Mirror Synchronization” on page 8-119
- “Suspend Remote Volume Mirror” on page 8-125

Session Command

- “Set Session” on page 8-76

Snapshot Commands

“Create Snapshot Volume” on page 8-28

“Re-create Snapshot” on page 8-51

“Set Snapshot Volume” on page 8-77

“Stop Snapshot” on page 8-122

Storage Array Commands

“Activate Storage Array Firmware” on page 8-7

“Autoconfigure Storage Array” on page 8-8

“Autoconfigure Storage Array Hot Spares” on page 8-10

“Clear Storage Array Configuration” on page 8-12

“Clear Storage Array Event Log” on page 8-13

“Clear Storage Array Firmware Pending Area” on page 8-13

“Disable Storage Array Feature” on page 8-39

“Download Storage Array Drive Firmware” on page 8-41

“Download Storage Array Firmware/NVSRAM” on page 8-43

“Download Storage Array NVSRAM” on page 8-44

“Enable Storage Array Feature Key” on page 8-45

“Reset Storage Array Battery Install Date” on page 8-56

“Reset Storage Array RLS Baseline” on page 8-57

“Reset Storage Array Volume Distribution” on page 8-57

“Save Storage Array Configuration” on page 8-62

“Save Storage Array Events” on page 8-63

“Save Storage Array Performance Statistics” on page 8-64

“Save Storage Array RLS Counts” on page 8-64

“Save Storage Array State Capture” on page 8-65

“Save Storage Array Support Data” on page 8-65

“Set Storage Array” on page 8-79

“Set Storage Array Time” on page 8-81

.....

- “Set Storage Array Tray Positions” on page 8-81
- “Show Storage Array Auto Configure” on page 8-102
- “Show Storage Array” on page 8-99
- “Show Storage Array Host Topology” on page 8-104
- “Show Storage Array LUN Mappings” on page 8-105
- “Show Storage Array Unreadable Sectors” on page 8-107
- “Start Storage Array Locate” on page 8-119
- “Stop Storage Array Drive Firmware Download” on page 8-123
- “Stop Storage Array Locate” on page 8-123

Module Commands

- “Download Environmental Card Firmware” on page 8-41
- “Start Tray Locate” on page 8-120
- “Stop Tray Locate” on page 8-124

Uncategorized Command

- “Show String” on page 8-108

Volume Commands

- “Check Volume Parity” on page 8-11
- “Clear Volume Reservations” on page 8-14
- “Clear Volume Unreadable Sectors” on page 8-14
- “Create RAID Volume (Automatic Drive Select)” on page 8-18
- “Create RAID Volume (Manual Drive Select)” on page 8-23
- “Create RAID Volume (Free Capacity Base Select)” on page 8-20
- “Delete Volume” on page 8-35
- “Recover RAID Volume” on page 8-47
- “Remove Volume LUN Mapping” on page 8-54
- “Repair Volume Parity” on page 8-55

-
- “Set Volume” on page 8-82
 - “Show Volume” on page 8-108
 - “Show Volume Action Progress” on page 8-110
 - “Show Volume Performance Statistics” on page 8-114
 - “Show Volume Reservations” on page 8-116
 - “Start Volume Initialization” on page 8-121

Volume Copy Commands

- “Create Volume Copy” on page 8-31
- “Recopy Volume Copy” on page 8-46
- “Remove Volume Copy” on page 8-53
- “Set Volume Copy” on page 8-87
- “Show Volume Copy” on page 8-111
- “Show Volume Copy Source Candidates” on page 8-112
- “Show Volume Copy Target Candidates” on page 8-113
- “Stop Volume Copy” on page 8-124

Volume Group Commands

- “Delete Volume Group” on page 8-36
- “Revive Volume Group” on page 8-60
- “Set Volume Group” on page 8-88
- “Show Volume Group” on page 8-113
- “Start Volume Group Defragment” on page 8-120

.....

Appendix A

Examples of Information Returned by Show Commands

This appendix provides examples of information returned by the show commands. The purpose of these example is to show the type of information and the information detail. This information is useful in determining the components, features, and identifiers you may need when configuring or maintaining a storage array.

Show Storage Array

The `show storageArray` command returns information about the components and features in a storage array. If you run the command with the `profile` parameter, the command returns information in the form shown by this example. This information is the most detailed report you can receive about the storage array. After you have configured a storage array, save the configuration description to a file as a reference.

```
Storage array profile
PROFILE FOR STORAGE ARRAY: example9 (x/xx/xx 1:06:05 PM)

SUMMARY-----
Number of controllers: 1
Number of volume groups: 1
Total number of volumes (includes an access volume): 5 of 2048 used
  Number of standard volumes: 2
  Number of access volumes: 1
  Number of mirror repositories: 2
  Number of snapshot repositories: 0
Number of snapshot volumes: 0
Number of mirrored pairs: 0 of 64 used
Number of copies: 0
Number of drives: 9
Supported drive types: Fibre (9)
Total hot spare drives: 0
  Standby: 0
  In use: 0
Access volume: LUN 7 (see Mappings section for details)
Default host type: Linux (Host type index 6)
Current configuration
  Firmware version: PkgInfo 96.10.21.00
  NVSRAM version: N4884-610800-001
Pending configuration
  Staged firmware download supported?: No
  Firmware version: Not applicable
  NVSRAM version: Not applicable
  Transferred on: Not applicable
NVSRAM configured for batteries?: Yes
Start cache flushing at (in percentage): 80
Stop cache flushing at (in percentage): 80
Cache block size (in KB): 4
Media scan frequency (in days): Disabled
Failover alert delay (in minutes): 5
Feature enable identifier: 1234567891011121314151617181ABCD
```

.....

CONTROLLERS-----

Number of controllers: 1

Controller in Slot A

Status: Online

Current configuration

Firmware version: 96.10.21.00

Appware version: 96.10.21.00

Bootware version: 96.10.21.00

NVSRAM version: N4884-610800-001

Pending configuration

Firmware version: Not applicable

Appware version: Not applicable

Bootware version: Not applicable

NVSRAM version: Not applicable

Transferred on: Not applicable

Board ID: 4884

Product ID: INF-01-00

Product revision: 9610

Serial number: 1T14148766

Date of manufacture: October 14, 2001

Cache/processor size (MB): 1024/128

Date/Time: Fri Feb 13 13:06:05 MST 2004

Associated Volumes (* = Preferred Owner):

1*, CTL 0 Mirror Repository*, Mirror Repository 1*, Primary_Volume*

Ethernet port: 1

MAC address: 00:a0:b8:0c:c3:f5

Host name: ausctlr9

Network configuration: Static

IP address: 172.22.4.249

Subnet mask: 255.255.255.0

Gateway: 172.22.4.1

Remote login: Enabled

Drive interface: Fibre

Channel: 1

Current ID: 125/0x1

Maximum data rate: 2 Gbps

Current data rate: 1 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 2

Current ID: 125/0x1

Maximum data rate: 2 Gbps

Current data rate: 1 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 3

Current ID: 125/0x1

Maximum data rate: 2 Gbps

Current data rate: 1 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 4

Current ID: 125/0x1

Maximum data rate: 2 Gbps

Current data rate: 1 Gbps

Data rate control: Switch

Link status: Up

```

.....

Host interface: Fibre
Port: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x011100
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:2c:00:a0:b8:0c:c3:f6
World-wide node name: 20:2c:00:a0:b8:0c:c3:f5
Part type: HPFC-5200    revision 10
Host interface: Fibre
Port: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x011000
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:2c:00:a0:b8:0c:c3:f7
World-wide node name: 20:2c:00:a0:b8:0c:c3:f5
Part type: HPFC-5200    revision 10

```

VOLUME GROUPS-----

Number of volume groups: 1

Volume Group 1 (RAID 5)

```

Status: Online
Drive type: Fibre Channel
Tray loss protection: No
Current owner: Controller in slot A
Associated volumes and free capacities:
  CTL 0 Mirror Repository (0.125 GB), Mirror Repository 1 (0.125 GB),
  Primary_Volume (5 GB), 1 (1 GB), Free Capacity (129.113 GB)
Associated drives (in piece order):
  Drive at Tray 0, Slot 1
  Drive at Tray 0, Slot 2
  Drive at Tray 0, Slot 3
  Drive at Tray 0, Slot 5
  Drive at Tray 0, Slot 6
  Drive at Tray 0, Slot 7
  Drive at Tray 0, Slot 10
  Drive at Tray 0, Slot 12
  Drive at Tray 0, Slot 14

```

STANDARD VOLUMES-----

SUMMARY

Number of standard volumes: 2
See other Volumes sub-tabs for premium feature information.

NAME	STATUS	CAPACITY	RAID LEVEL	VOLUME GROUP
1	Optimal	1 GB	5	1
Primary_Volume	Optimal	5 GB	5	1

DETAILS

Volume name: 1
Volume ID: 60:0a:0b:80:00:0c:c3:f5:00:00:00:23:40:22:86:ea
Subsystem ID (SSID): 3
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 1 GB
RAID level: 5
Segment size: 64 KB
Modification priority: High
Associated volume group: 1
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled

Volume name: Primary_Volume
Volume ID: 60:0a:0b:80:00:0c:c3:f5:00:00:00:1c:40:22:45:ca
Subsystem ID (SSID): 2
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 5 GB
RAID level: 5
Segment size: 64 KB
Modification priority: High
Associated volume group: 1
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled

.....

SNAPSHOT REPOSITORY VOLUMES-----
 Number of snapshot repositories: 0

MIRROR REPOSITORY VOLUMES-----
 Number of mirror repositories: 0

SNAPSHOT VOLUMES-----
 Number of snapshot volumes: 0

MIRRORED PAIRS-----

SUMMARY
 Number of mirrored pairs: 0 of 64 used
 See also Standard sub-tab for standard volume properties

COPIES-----

SUMMARY
 Number of copies: 0
 See also Standard sub-tab for standard volume properties

MISSING VOLUMES-----
 Number of missing volumes: 0
 See other volumes sub-tabs for premium feature information

DRIVES-----

SUMMARY
 Number of drives: 9
 Supported drive types: Fibre (9)

BASIC:

TRAY, SLOT	STATUS	CAPACITY	CURRENT DATA RATE	PRODUCT ID	FIRMWARE VERSION
0, 1	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 2	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 3	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 5	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 6	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 7	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 10	Optimal	16.959 GB	1 Gbps	ST318451FC	F280
0, 12	Optimal	16.959 GB	1 Gbps	ST318451FC	F288
0, 14	Optimal	16.959 GB	1 Gbps	ST318451FC	F288

DRIVE CHANNELS:

TRAY, SLOT	PREFERRED CHANNEL	REDUNDANT CHANNEL
0, 1	4	2
0, 2	2	4
0, 3	4	2
0, 5	4	2
0, 6	2	4
0, 7	4	2
0, 10	2	4
0, 12	2	4
0, 14	2	4

.....

HOT SPARE COVERAGE:
The following volume groups are not protected: 1

Total hot spare drives: 0
Standby: 0
In use: 0

DETAILS

Drive at Tray 0, Slot 1
Drive port: 1, Channel: 4, ID: 0/0xEF
Drive port: 2, Channel: 2, ID: 0/0xEF
Drive path redundancy: OK
Status: Optimal
Raw capacity: 16.959 GB
Usable capacity: 16.92 GB
Current data rate: 1 Gbps
Product ID: ST318451FC
Firmware version: F28D
Serial number: 3CC052RK00007116BMW7
Vendor: SEAGATE
Date of manufacture: July 19, 2001
World-wide name: 20:00:00:20:37:f5:b3:b6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 1

Drive at Tray 0, Slot 2
Drive port: 1, Channel: 2, ID: 1/0xE8
Drive port: 2, Channel: 4, ID: 1/0xE8
Drive path redundancy: OK
Status: Optimal
Raw capacity: 16.959 GB
Usable capacity: 16.92 GB
Current data rate: 1 Gbps
Product ID: ST318451FC
Firmware version: F28D
Serial number: 3CC0Q06K0000720291J9
Vendor: SEAGATE
Date of manufacture: July 18, 2001
World-wide name: 20:00:00:20:37:f5:ac:cd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 1

Drive at Tray 0, Slot 3
Drive port: 1, Channel: 4, ID: 2/0xE4
Drive port: 2, Channel: 2, ID: 2/0xE4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 16.959 GB
Usable capacity: 16.92 GB
Current data rate: 1 Gbps
Product ID: ST318451FC
Firmware version: F28D
Serial number: 3CC052MA00007116DB89
Vendor: SEAGATE
Date of manufacture: July 19, 2001
World-wide name: 20:00:00:20:37:f5:b3:fc
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 1

```

.....

Drive at Tray 0, Slot 5
  Drive port: 1, Channel: 4, ID: 4/0xE1
  Drive port: 2, Channel: 2, ID: 4/0xE1
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F280
  Serial number: 3CC0526N00007143A10K
  Vendor: SEAGATE
  Date of manufacture: July 19, 2001
  World-wide name: 20:00:00:20:37:f5:b4:5b
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

Drive at Tray 0, Slot 6
  Drive port: 1, Channel: 2, ID: 5/0xE0
  Drive port: 2, Channel: 4, ID: 5/0xE0
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F280
  Serial number: 3CC03R5800007112JU2D
  Vendor: SEAGATE
  Date of manufacture: July 19, 2001
  World-wide name: 20:00:00:20:37:f5:b3:06
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

Drive at Tray 0, Slot 7
  Drive port: 1, Channel: 4, ID: 6/0xDC
  Drive port: 2, Channel: 2, ID: 6/0xDC
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F280
  Serial number: 3CC052J400007116BSJV
  Vendor: SEAGATE
  Date of manufacture: July 19, 2001
  World-wide name: 20:00:00:20:37:f5:b3:0b
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

```

```

.....

Drive at Tray 0, Slot 10
  Drive port: 1, Channel: 2, ID: 96/0x3A
  Drive port: 2, Channel: 4, ID: 96/0x3A
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F28D
  Serial number: 3CC0P5Q80000711552CZ
  Vendor: SEAGATE
  Date of manufacture: June 30, 2001
  World-wide name: 20:00:00:20:37:f5:69:50
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

Drive at Tray 0, Slot 12
  Drive port: 1, Channel: 2, ID: 72/0x67
  Drive port: 2, Channel: 4, ID: 72/0x67
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F28B
  Serial number: 3CC0P7TS00007116DL9D
  Vendor: LSILOGIC
  Date of manufacture: Not available
  World-wide name: 20:00:00:20:37:f5:68:76
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

Drive at Tray 0, Slot 14
  Drive port: 1, Channel: 2, ID: 104/0x2E
  Drive port: 2, Channel: 4, ID: 104/0x2E
  Drive path redundancy: OK
  Status: Optimal
  Raw capacity: 16.959 GB
  Usable capacity: 16.92 GB
  Current data rate: 1 Gbps
  Product ID: ST318451FC
  Firmware version: F28B
  Serial number: 3CC03STH00007112JNM4
  Vendor: LSILOGIC
  Date of manufacture: Not available
  World-wide name: 20:00:00:20:37:f5:b3:69
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: Volume group 1

```

```

.....

DRIVE CHANNEL INFORMATION - Channel 1

SUMMARY-----
Channel 1 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:37 AM
Sample period (days, hh:mm:ss): 4 days, 02:06:52
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 3
Total I/O count: 11

DRIVE CHANNEL INFORMATION - Channel 2

SUMMARY-----
Channel 2 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 4 days, 02:06:47
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 2
Total I/O count: 239075

DRIVE CHANNEL INFORMATION - Channel 3

SUMMARY-----
Channel 3 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 4 days, 02:06:47
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 3
Total I/O count: 11

DRIVE CHANNEL INFORMATION - Channel 4

SUMMARY-----
Channel 4 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 4 days, 02:06:47
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 2
Total I/O count: 45807

```

```

.....

TRAYS-----
Controller Tray Overall Component Information
Drive technology: Unknown
Minihub data rate mismatch: No
Fan canister: Optimal
Fan canister: Optimal
Battery status: Optimal
  Age: 0 day(s)
  Days until replacement: 719 day(s)
Power supply canister
  Status: Optimal
Power supply canister
  Status: Optimal
Temperature: Optimal
Host mini-hub canister
  Status: Optimal
  Location: Controller A - Port 1
  Serial number: SN 1T13742151
  Part number: PN 348-0043290
  Vendor: VN LSILOGIC
  Date of manufacture: Not available
SFP
  Status: Optimal
  Location: In connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P215NY
  Serial number: 21P7053000M6R
  Vendor: IBM
  Date of manufacture: July 1, 2001
SFP
  Status: Optimal
  Location: Out connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P215NY
  Serial number: 21P7053000M54
  Vendor: IBM
  Date of manufacture: July 1, 2001

```

```

.....

Host mini-hub canister
Status: Optimal
Location: Controller A - Port 2
Serial number: SN 1T13742219
Part number: PN 348-0043290
Vendor: VN LSILOGIC
Date of manufacture: Not available
SFP
  Status: Optimal
  Location: In connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M3Z
  Vendor: IBM
  Date of manufacture: July 1, 2001
SFP
  Status: Optimal
  Location: Out connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M3J
  Vendor: IBM
  Date of manufacture: July 1, 2001
Drive mini-hub canister
Status: Optimal
Location: Channel 1
Serial number: SN 1T13438343
Part number: PN 348-0043290
Vendor: VN LSILOGIC
Date of manufacture: Not available
SFP
  Status: Optimal
  Location: Out connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M6Z
  Vendor: IBM
  Date of manufacture: July 1, 2001

```

```

.....

Drive mini-hub canister
Status: Optimal
Location: Channel 2
Serial number: SN 1T13844441
Part number: PN 19K1270
Vendor: VN IBM
Date of manufacture: Not available
SFP
  Status: Optimal
  Location: In connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M4E
  Vendor: IBM
  Date of manufacture: July 1, 2001
SFP
  Status: Optimal
  Location: Out connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M40
  Vendor: IBM
  Date of manufacture: July 1, 2001
Drive mini-hub canister
Status: Optimal
Location: Channel 3
Serial number: SN 1T13844373
Part number: PN 348-0043290
Vendor: VN LSILOGIC
Date of manufacture: Not available
SFP
  Status: Optimal
  Location: In connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000M4R
  Vendor: IBM
  Date of manufacture: July 1, 2001
SFP
  Status: Optimal
  Location: Out connection
  Supported data rate(s): 1 Gbps, 2 Gbps
  Link length: Intermediate
  Connector: LC
  Transmitter type: Shortwave Laser w/o OFC
  Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
  IEEE company ID: 08 00 5a
  Revision: AA10
  Part number: IBM42P21SNY
  Serial number: 21P7053000LXS
  Vendor: IBM
  Date of manufacture: July 1, 2001

```

.....

Drive mini-hub canister
Status: Optimal
Location: Channel 4
Serial number: SN 1T13844386
Part number: PN 348-0043290
Vendor: VN LSILOGIC
Date of manufacture: Not available

SFP
Status: Optimal
Location: In connection
Supported data rate(s): 1 Gbps, 2 Gbps
Link length: Intermediate
Connector: LC
Transmitter type: Shortwave Laser w/o OFC
Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
IEEE company ID: 08 00 5a
Revision: AA10
Part number: IBM42P21SNY
Serial number: 21P7053000M6E
Vendor: IBM
Date of manufacture: July 1, 2001

SFP
Status: Optimal
Location: Out connection
Supported data rate(s): 1 Gbps, 2 Gbps
Link length: Intermediate
Connector: LC
Transmitter type: Shortwave Laser w/o OFC
Transmission media: TM Multi-mode 50m(M5) TM Multi-mode 62.5m(M6)
IEEE company ID: 08 00 5a
Revision: AA10
Part number: IBM42P21SNY
Serial number: 21P7053000M57
Vendor: IBM
Date of manufacture: July 1, 2001

.....

Drive Tray 0 Overall Component Information

Drive technology: Fibre channel
Part number: PN 348-0042807
Serial number: SN 0146001584
Vendor: VN LSILOGIC
Date of manufacture: November 1, 2001
Tray path redundancy: OK
Fan canister: Optimal
Fan canister: Optimal
Power supply canister
Status: Optimal
Power supply canister
Status: Optimal
Temperature: Optimal
Temperature: Optimal

ESM card

Status: Optimal
Firmware version: 9163
Maximum data rate: 1 Gbps
Current data rate: 1 Gbps
Location: A (left canister)
Card communication: OK
Product ID: SYM2500-FC
Part number: PN 348-0043832
Serial number: SN 1T22941398
Vendor: LSILOGIC
Date of manufacture: July 1, 2002

GBIC

Status: Optimal
Location: In connection
Type: Optical Shortwave

ESM card

Status: Optimal
Firmware version: 9163
Maximum data rate: 1 Gbps
Current data rate: 1 Gbps
Location: B (right canister)
Card communication: OK
Product ID: SYM2500-FC
Part number: PN 348-0043832
Serial number: SN 1T13945863
Vendor: LSILOGIC
Date of manufacture: October 1, 2001

GBIC

Status: Optimal
Location: In connection
Type: Optical Shortwave

```

.....
MAPPINGS (SANshare STORAGE PARTITIONING - ENABLED (0 of 64 used))-----
Volume-To-LUN Mappings
  Access Volume, LUN 7, Default Group

Topology
  [Undefined Host Ports]
    10:00:00:a0:b8:04:29:3f
    21:00:00:e0:8b:01:34:c1
    21:01:00:e0:8b:22:80:38

  [Default Group]

NVRAM Host Type Internal Definitions
Index 0
  Name: windows NT Non-Clustered (SP5 or higher)
  AVT status: Disabled
Index 1
  Name: windows 2000/Server 2003 Non-Clustered
  AVT status: Disabled
Index 2
  Name: Solaris
  AVT status: Disabled
Index 3
  Name: HP-UX
  AVT status: Enabled
Index 4
  Name: AIX
  AVT status: Disabled
Index 5
  Name: Irix
  AVT status: Disabled
Index 6 (DEFAULT)
  Name: Linux
  AVT status: Disabled
Index 7
  Name: Windows NT Clustered (SP5 or higher)
  AVT status: Disabled
Index 8
  Name: Windows 2000/Server 2003 Clustered
  AVT status: Disabled
Index 9
  Name: Netware Non-Failover
  AVT status: Enabled
Index 10
  Name: PTX
  AVT status: Enabled
Index 11
  Name: Netware Failover
  AVT status: Enabled
Index 12
  Name: Solaris (with Veritas DMP)
  AVT status: Enabled

```

Show Controller NVSRAM

The `show controller NVSRAM` command returns a table of the values in the controller NVSRAM similar to that shown in this example. With the information from the table, you can modify the contents of the NVSRAM using the `set controller` command. This example shows information for a controller in slot A in a command module. You can produce a similar table for a controller in slot B, or you can produce a table for both controllers.

```

Controller "a":
Region 232:
  0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 233:
  0000: 0000 7f8e 0000 1384 ff01 0000 0000 0000 ...?...?.....
  0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0020: 0000 0308 8e08 8108 0000 0000 0000 0408 ....??.?.....
  0030: 0220 0220 8920 8820 8d00 0900 0308 0000 ....??.?.....
  0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0050: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 234:
  0000: 1020 3040 5060 0000 0000 0000 0000 0000 ..0.P.....
  0010: 1121 3141 5161 0000 0000 0000 0000 0000 ..1AQa.....
  0020: 1222 3242 5262 0000 0000 0000 0000 0000 ..2BRb.....
  0030: 1323 3343 5363 0000 0000 0000 0000 0000 ..3CSc.....
  0040: 1424 3444 5464 0000 0000 0000 0000 0000 ..4DTd.....
  0050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0070: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 236:
  0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0070: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 237:
  0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 238:
  0000: 0000 ca32 0000 0000 0050 0600 0000 0000 ..?2.....P.....
  0010: 0000 0000 0000 0000 0000 f001 0000 8480 0000 .....???.
  0020: 0000 0000 0000 0000 0000 0c04 008a 0000 0000 .....?.....
  0030: 0020 8741 1300 0000 0f00 1400 0000 0000 ..?A.....

```



```

.....

Region 242:
0000: 4c4e 5800 0000 0000 0000 0000 0000 0000 LNX.....
0010: 0120 0000 0100 0000 0001 0000 0000 0000 .....
0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 574e 5443 4c53 5035 0000 0000 0000 0000 WNTCLSP5.....
0010: 0000 0000 0100 0000 0001 0001 0000 0000 .....
0020: 0100 0000 0000 0002 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 5732 4b4e 4554 434c 0000 0000 0000 0000 W2KNETCL.....
0010: 0020 0000 0100 0000 0001 0001 0000 0000 .....
0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 4e57 5200 0000 0000 0000 0000 0000 0000 NWR.....
0010: 007f 0000 0100 2000 0000 0000 0000 0000 .....
0020: 0100 0100 0100 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 5054 5800 0000 0000 0000 0000 0000 0000 PTX.....
0010: 0120 0000 0100 0000 0001 0000 0000 0000 .....
0020: 0100 0000 0100 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 4e57 5246 4f00 0000 0000 0000 0000 0000 NWRFO.....
0010: 007f 0000 0100 0000 0000 0000 0000 0000 .....
0020: 0100 0100 0100 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 534f 4c41 5654 0000 0000 0000 0000 0000 SOLAVT.....
0010: 0120 0000 0100 0000 0001 0100 0000 0000 .....
0020: 0000 0001 0180 0000 0000 0000 0000 0000 .....?.....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

Region 242:
0000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

Show Drive

The show drive command returns a list of information about the drives in a storage array.

DRIVES-----

SUMMARY

Number of drives: 14
Supported drive types: Fibre (14)

BASIC:

TRAY, SLOT	STATUS	CAPACITY	CURRENT DATA RATE	PRODUCT ID	FIRMWARE VERSION
1, 1	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 2	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 3	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 4	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 5	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 6	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 7	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 8	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 9	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 10	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 11	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 12	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 13	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03
1, 14	Optimal	68.366 GB	2 Gbps	ST373453FC	MS03

DRIVE CHANNELS:

TRAY, SLOT	PREFERRED CHANNEL	REDUNDANT CHANNEL
1, 1	3	Drive path redundancy: Lost
1, 2	3	Drive path redundancy: Lost
1, 3	3	Drive path redundancy: Lost
1, 4	3	Drive path redundancy: Lost
1, 5	3	Drive path redundancy: Lost
1, 6	3	Drive path redundancy: Lost
1, 7	3	Drive path redundancy: Lost
1, 8	3	Drive path redundancy: Lost
1, 9	3	Drive path redundancy: Lost
1, 10	3	Drive path redundancy: Lost
1, 11	3	Drive path redundancy: Lost
1, 12	3	Drive path redundancy: Lost
1, 13	3	Drive path redundancy: Lost
1, 14	3	Drive path redundancy: Lost

HOT SPARE COVERAGE:

The following volume groups are not protected: 3

Total hot spare drives: 2

Standby: 2

In use: 0

Standby drive at tray 1, slot 8 (Fibre, 68.366 GB)

Protects the following volume groups: 2, 1

.....

DETAILS

Drive at Tray 1, Slot 1
Drive port: 1, Channel: 3, ID: 8/0x09
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: M503
Serial number: 3HW0535L00007329LFD8
Vendor: SEAGATE
Date of manufacture: January 20, 2003
World-wide name: 20:00:00:04:cf:0b:c6:0d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 2

Drive at Tray 1, Slot 2
Drive port: 1, Channel: 3, ID: 9/0x06
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: M503
Serial number: 3HW0560500007329VWAP
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:dc:29
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 2

Drive at Tray 1, Slot 3
Drive port: 1, Channel: 3, ID: 10/0x05
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: M503
Serial number: 3HW04LMK00007329DYZK
Vendor: SEAGATE
Date of manufacture: January 17, 2003
World-wide name: 20:00:00:04:cf:0b:b1:f6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Unassigned
Associated volume group: Volume group None

.....

Drive at Tray 1, Slot 4
Drive port: 1, Channel: 3, ID: 11/0x04
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04N2T00007329DZGZ
Vendor: SEAGATE
Date of manufacture: January 17, 2003
World-wide name: 20:00:00:04:cf:0b:b1:81
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 3

Drive at Tray 1, Slot 5
Drive port: 1, Channel: 3, ID: 12/0x03
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04YN500007329JQYG
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:db:2c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Unassigned
Associated volume group: Volume group None

Drive at Tray 1, Slot 6
Drive port: 1, Channel: 3, ID: 13/0x02
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW056N3000073282C57
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:db:66
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 2

.....

Drive at Tray 1, Slot 7
Drive port: 1, Channel: 3, ID: 14/0xD1
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04XKQ00007329VUR8
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:db:a1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: volume group 2

Drive at Tray 1, Slot 8
Drive port: 1, Channel: 3, ID: 15/0xCE
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04QL300007328H6BU
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:db:6f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Hot spare standby
Associated volume group: volume group None

Drive at Tray 1, Slot 9
Drive port: 1, Channel: 3, ID: 81/0x54
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04W5E00007329EQX2
Vendor: SEAGATE
Date of manufacture: January 18, 2003
World-wide name: 20:00:00:04:cf:0b:b2:5b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Unassigned
Associated volume group: volume group None

.....

Drive at Tray 1, Slot 10
Drive port: 1, Channel: 3, ID: 97/0x39
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04V5P00007329DZPH
Vendor: SEAGATE
Date of manufacture: January 17, 2003
World-wide name: 20:00:00:04:cf:0b:b1:97
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 2

Drive at Tray 1, Slot 11
Drive port: 1, Channel: 3, ID: 65/0x71
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW056N700007329T7EG
Vendor: SEAGATE
Date of manufacture: January 23, 2003
World-wide name: 20:00:00:04:cf:0b:db:69
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 2

Drive at Tray 1, Slot 12
Drive port: 1, Channel: 3, ID: 73/0x66
Drive path redundancy: Lost
Status: Optimal
Raw capacity: 68.366 GB
Usable capacity: 67.866 GB
Current data rate: 2 Gbps
Product ID: ST373453FC
Firmware version: MS03
Serial number: 3HW04DL400007329ERF1
Vendor: SEAGATE
Date of manufacture: January 17, 2003
World-wide name: 20:00:00:04:cf:0b:b1:84
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated volume group: Volume group 1

.....

```
Drive at Tray 1, Slot 13
  Drive port: 1, Channel: 3, ID: 89/0x4A
  Drive path redundancy: Lost
  Status: Optimal
  Raw capacity: 68.366 GB
  Usable capacity: 67.866 GB
  Current data rate: 2 Gbps
  Product ID: ST373453FC
  Firmware version: MS03
  Serial number: 3HW04DJT00007329DZJF
  Vendor: SEAGATE
  Date of manufacture: January 18, 2003
  World-wide name: 20:00:00:04:cf:0b:b0:ff
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: volume group 1
```

```
Drive at Tray 1, Slot 14
  Drive port: 1, Channel: 3, ID: 105/0x2D
  Drive path redundancy: Lost
  Status: Optimal
  Raw capacity: 68.366 GB
  Usable capacity: 67.866 GB
  Current data rate: 2 Gbps
  Product ID: ST373453FC
  Firmware version: MS03
  Serial number: 3HW04VAK00007329ER3P
  Vendor: SEAGATE
  Date of manufacture: January 17, 2003
  World-wide name: 20:00:00:04:cf:0b:b1:6e
  Drive type: Fibre Channel
  Speed: 15015 RPM
  Mode: Assigned
  Associated volume group: volume group 1
```

Show Drive Channel Status

The `show drivechannel stat` command returns a table listing information about the drive channels in a storage array. Use this information to determine how well the channels are running and errors that might be occurring on the channels.

```
DRIVE CHANNEL INFORMATION - Channel 1
SUMMARY-----
Channel 1 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:37 AM
Sample period (days, hh:mm:ss): 8 days, 22:44:25
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 3
Total I/O count: 22

DRIVE CHANNEL INFORMATION - Channel 2
SUMMARY-----
Channel 2 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 8 days, 22:44:20
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 2
Total I/O count: 412340

DRIVE CHANNEL INFORMATION - Channel 3
SUMMARY-----
Channel 3 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 8 days, 22:44:20
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 3
Total I/O count: 22

DRIVE CHANNEL INFORMATION - Channel 4
SUMMARY-----
Channel 4 status: Optimal
Controller A link status: Up

CUMULATIVE ERROR COUNTS-----
Controller A Cumulative Error Counts
Baseline time set: 2/9/04 10:46:42 AM
Sample period (days, hh:mm:ss): 8 days, 22:44:20
Controller detected errors: 0
Drive detected errors: 0
Timeout errors: 0
Link down errors: 2
Total I/O count: 162911
```

Show Volumes

The show volume command returns a list of information about the volumes in a storage array.

```
STANDARD VOLUMES-----
```

SUMMARY

```
Number of standard volumes: 4
See other Volumes sub-tabs for premium feature information.
```

NAME	STATUS	CAPACITY	RAID LEVEL	VOLUME GROUP
1	Optimal	1 GB	5	1
2	Optimal	1 GB	0	3
RAID1testusingGUI	Optimal	1 GB	1	2
Test	Optimal	1 GB	0	3

DETAILS

```
Volume name: 1
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:1f:40:04:0b:1a
Subsystem ID (SSID): 0
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 1 GB
RAID level: 5
Segment size: 64 KB
Modification priority: High
Associated volume group: 1
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled
```

```
Volume name: 2
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:36:40:2c:9b:70
Subsystem ID (SSID): 4
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 1 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated volume group: 3
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled
```

.....

```
Volume name: RAID1testusingGUI
Volume ID: 60:0a:0b:80:00:0f:21:8a:00:00:00:1b:40:12:8f:45
Subsystem ID (SSID): 1
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 1 GB
RAID level: 1
Segment size: 64 KB
Modification priority: High
Associated volume group: 2
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled
```

```
Volume name: Test
Volume ID: 60:0a:0b:80:00:0f:21:8a:00:00:00:21:40:22:44:47
Subsystem ID (SSID): 3
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 1 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated volume group: 3
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Disabled
Media scan with redundancy check: Disabled
```

.....

SNAPSHOT REPOSITORY VOLUMES-----

SUMMARY

Number of snapshot repositories: 1

Number of snapshot repositories: 1

NAME	CAPACITY	USAGE(%)	THRESHOLD	WARNING	FULL POLICY
1R1	0		50% full		Fail snapshot volume

DETAILS

Snapshot repository name: 1R1
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:27:40:14:ed:4a
Subsystem ID (SSID): 2
Status: Optimal
Drive type: Fibre Channel
Tray loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 0.2 GB
RAID level: 5
Segment size: 64 KB
Modification priority: High
Associated volume group: 1
Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Disabled
Write cache with mirroring: Enabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 0
Enable background media scan: Disabled
Media scan with redundancy check: Disabled
Capacity usage (%): 0
Notify when capacity reaches: 50% full
Snapshot repository full policy: Fail snapshot volume
Associated base volume (standard): 1
Associated snapshot volume: 1S1

.....
MIRROR REPOSITORY VOLUMES-----
Number of mirror repositories: 0

SNAPSHOT VOLUMES-----

SUMMARY
Number of snapshot volumes: 1

NAME	STATUS	CREATION TIMESTAMP
1S1	Optimal	1/26/04 9:34 AM

DETAILS

Snapshot volume name: 1S1
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:2a:40:14:ed:4c
Creation timestamp: 1/26/04 9:34 AM
Capacity: 1 GB
Status: Optimal
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Associated base volume (standard): 1
Associated snapshot repository volume: 1R1

COPIES-----

SUMMARY
Number of copies: 2
See also standard sub-tab for standard volume properties
S = Source volume
T = Target volume

COPY PAIR	STATUS	COMPLETION TIMESTAMP
1 (S), RAID1testusingGUI (T)	Completed	2/5/04 9:19:34 AM
Test (S), 2 (T)	Completed	2/13/04 8:46:57 AM

DETAILS

Copy pair: 1 and RAID1testusingGUI
Copy status: Completed
Start timestamp: 2/5/04 9:17:09 AM
Completion timestamp: 2/5/04 9:19:34 AM
Copy priority: Medium
Source volume: 1
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:1f:40:04:0b:1a
Target volume: RAID1testusingGUI
Volume ID: 60:0a:0b:80:00:0f:21:8a:00:00:00:1b:40:12:8f:45
Read-only: Enabled

Copy pair: Test and 2
Copy status: Completed
Start timestamp: 2/13/04 8:46:14 AM
Completion timestamp: 2/13/04 8:46:57 AM
Copy priority: Highest
Source volume: Test
Volume ID: 60:0a:0b:80:00:0f:21:8a:00:00:00:21:40:22:44:47
Target volume: 2
Volume ID: 60:0a:0b:80:00:0f:21:a8:00:00:00:36:40:2c:9b:70
Read-only: Enabled

Appendix B

Example Script Files

This appendix provides example scripts for configuring a storage array. The purpose of these example is to show the how the script commands appear in a complete script file. Also, users can copy these scripts and modify them to create a configuration unique to their storage array.

You can create a script file in two ways:

- Using the `save storageArray` configuration command
- Writing a script

By using the `save storageArray` configuration command, you can create a file that you can use to copy an existing configuration from one storage array to others. You can also use this file to restore an existing configuration that has become corrupted. You can also copy an existing file to serve as pattern from which you create a new script file by modifying portions of the original file. The default file extension is `.scr`.

You can create a new script file using a text editor such as Microsoft Notepad. The maximum line length is 256 characters. The command syntax must conform to the usage guidelines on page 2-10 and the command formatting rules on page 8-2. When you create a new script file, you can use any file name and extension that will run on the host operating system.

To run a script file from the command line, enter:

```
c:\...\sm9\client>smcli 123.45.67.89 -f scriptfile.scr;
```

Configuration Script Example 1

This example creates a new volume using the `create volume` command in the free space of a volume group.

```
Show "Create RAID 5 Volume 7 on existing Volume Group 1";
```

```
//Create volume on volume group created by the create volume drives command
```

```
//Note: For volume groups that use all available capacity, the last volume on the group is created using all remaining capacity by omitting the capacity=volume creation parameter
```

```
create volume volumeGroup=1 RAIDLevel=5 userLabel="7" owner=A  
segmentSize=16 readAheadMultiplier=256 capacity=2GB;
```

```
show "Setting additional attributes for volume 7";  
//Configuration settings that cannot be set during volume creation  
set volume["7"] cacheFlushModifier=10;  
set volume["7"] cacheWithoutBatteryEnabled=false;  
set volume["7"] mirrorEnabled=true;  
set volume["7"] readCacheEnabled=true;  
set volume["7"] writeCacheEnabled=true;  
set volume["7"] mediaScanEnabled=false;  
set volume["7"] redundancyCheckEnabled=false;  
set volume["7"] modificationPriority=high;
```

This example shows blank lines between the lines beginning with “Show,” “Create,” “// Note,” and “create.” The blank lines are included in this example only for clarity. Each command is actually written on one line in the script file; however, the size of this page has caused the command text to wrap. You may want to include blank lines in your script files to separate blocks of commands or make a comment more outstanding. You can do this by entering two forward slashes (`//`), which causes the script engine to treat the line as a comment.

The first line of text is the `show string` command. This command displays text bounded by double quotes (“ ”) on a monitor screen when the script file runs. In this example the text “Create RAID 5 Volume 7 on existing Volume Group 1” serves as a title describing the expected results of running this script file.

The line beginning “//Create” is a comment explaining that the purpose of this script file is to create a new volume using the `create volume` command on an existing volume group.

.....

The line beginning “//Note:” is a comment in the script file explaining that the size of the last volume created uses all available capacity because the `capacity` parameter is not used.

The command in this example creates a new volume in volume group 1. The volume will have a RAID level of 5. The volume name (user label) will be 7. (Note the double quotation marks around the 7. The double quotation marks define that the information in the marks is a label.) The new volume will be assigned to the controller in slot A in the command module. The segment size will be set to 16. The volume will have a read ahead multiplier value of 256. The volume capacity will be 2GB.

The general form of the command is

```
create volume volumeGroup=volumeGroupName
userLabel=volumeName [freeCapacityArea=freeCapacityValue]
[capacity=volumeCapacity | owner=(a | b) |
readAheadMultiplier=multiplierValue |
segmentSize=segmentSizeValue] [trayLossProtect=(TRUE |
FALSE) ]
```

The general form of the command shows the optional parameters in a different sequence than the optional parameters in the example command. You can enter optional parameters in any sequence. You must enter the required parameters in the sequence shown in the command descriptions.

The line “show “Setting additional attributes for volume 7”” is another example of using the `show string` command. The reason for placing this command here is to tell the user that the `create volume` command ran successfully and that properties that could not be set by the `create volume` command will now be set.

The `set volume` command parameters are shown on separate lines. You do not need to use separate lines for each parameter. You can enter more than one parameter with the `set volume` command by leaving a space between the parameters. For example:

```
set volume["7"] cacheFlushModifier=10 cacheWithoutBatteryEnabled=false
modificationPriority=high;
```

By using separate lines, however, you can more clearly see what parameters you are setting and the values to which you are setting the parameters. Blocking the parameters in this manner makes it easier to edit the file or copy specific parameter settings for use in another script file.

Configuration Script Example 2

This example creates a new volume using the `create volume` command with user defined physical disks in the storage array.

Show “Create RAID 3 Volume 2 on existing Volume Group 2”;

//This command creates the volume group and the initial volume on that group.

//Note: For volume groups that use all available capacity, the last volume on the group is created using all remaining capacity by omitting the `capacity=volume` creation parameter

```
create volume RAIDLevel=3 userLabel="2" drives=[0,1 0,6 1,7 1,3 2,3 2,6]
owner=B segmentSize=16 capacity=2GB;
```

```
show "Setting additional attributes for volume 7";
//Configuration settings that cannot be set during volume creation
set volume["7"] cacheFlushModifier=10;
set volume["7"] cacheWithoutBatteryEnabled=false;
set volume["7"] mirrorEnabled=true;
set volume["7"] readCacheEnabled=true;
set volume["7"] writeCacheEnabled=true;
set volume["7"] mediaScanEnabled=false;
set volume["7"] redundancyCheckEnabled=false;
set volume["7"] modificationPriority=high;
```

The command in this example, like the `create volume` command in the previous example, creates a new volume. The significant difference between these two examples is that this example shows how a user can define specific physical disks to include in the volume. To find out what physical disks are available in a storage array, run the `show storageArray profile` command.

The general form of the `create volume` command is:

```
create volume raidLevel=(0 | 1 | 3 | 5) userLabel=volumeName
drives=(trayID1,slotID1...trayIDn,slotIDn)
[capacity=volumeCapacity | owner=(a | b) |
readAheadMultiplier=multiplierValue |
segmentSize=segmentSizeValue] [trayLossProtect=(TRUE |
FALSE) ]
```

Appendix C

Deprecated Commands and Keywords

This appendix lists the commands, command formats, and keywords no longer supported by this level of software. The information is presented in the form of two tables. Table C-1 lists commands no longer supported in this level of software and the new commands that replace them. Table C-2 on page C-7 lists the keywords no longer supported in this level of software and the new keywords that replace them.

Table C-1 Deprecated Commands

Deprecated Command	New Command
<code>create volume (drive drives) [trayID1,slotID1...trayIDn,slotIDn]</code>	<code>create volume drives=(trayID1,slotID1...trayIDn,slotIDn)</code> The new syntax for specifying drives requires an equal sign (=) after the keyword “drives.”
<code>create volume driveCount [numberOfDrives]</code>	<code>create volume driveCount=numberOfDrives</code> The new syntax for specifying the number of drives requires an equal sign (=) after the keyword “driveCount.”
<code>create volume volumeGroup [numberOfDrives]</code>	<code>create volume volumeGroup=volumeGroupNumber</code> The new syntax for specifying the volume group number requires an equal sign (=) after the keyword “volumeGroup.”

Deprecated Command	New Command
<pre>download storageArray (firmwareFile NVSRAMFile)=filename</pre>	<pre>download drive [trayID,slotID]firmware file="filename"</pre> <pre>download storageArray firmware [, NVSRAM] file="filename" [, "NVSRAM-filename"] [downgrade=(TRUE FALSE)] [activateNow=(TRUE FALSE)]</pre> <p>The new version of the storage management software provides provide unique commands to perform these functions.</p>
<pre>set controller [(a b)] NVSRAMByte [nvram- offset]=(nvramByteSetting nvramBitSetting)</pre>	<pre>set controller [(a b)] globalNVSRAMByte [nvramOffset]=(nvramByteSett ing nvramBitSetting)</pre> <p>This new command provides additional parameters for setting NVSRAM values.</p>
<pre>show storageArray preferredVolumeOwners</pre>	<pre>show storageArray profile</pre> <p>This command with the profile parameter returns the preferred volume owner information.</p>
<pre>show volumes volume [userLabel]</pre>	<pre>show storageArray profile</pre> <p>This command with the profile parameter returns the volume information.</p>
<pre>download tray [0]</pre>	<pre>download allTrays firmware file="filename"</pre> <p>When downloading ESM firmware to all drive modules, in the previous command “all modules” was defined by entering [0]. The new command uses the allTrays parameter.</p>

Deprecated Command	New Command
<pre>capacity=<i>volumeCapacity</i> raidLevel=(0 1 3 5) userLabel=<i>volumeName</i> owner=(a b) usageHint=<i>usageHintSpec</i> segmentSize=<i>segmentSizeValue</i> readAheadMultiplier=<i>multiplier Value</i></pre>	<pre>capacity=<i>volumeCapacity</i> owner=(a b) readAheadMultiplier=<i>multiplier Value</i> segmentSize=<i>segmentSizeValue</i></pre> <p>In the create raid volume commands, the new commands uses fewer optional attribute value pairs.</p>
<pre>create mapping volume=<i>userLabel</i> logicalGroupNumber=<i>logicalGroup pNumber</i> [host hostGroup]=<i>hostName</i> <i>hostGroupName</i></pre>	<p>Use the set volume command to define volume LUN mapping. Refer to “Set Volume” on page 8-82.</p>
<pre>delete mapping volume=<i>userLabel</i> [host hostGroup]=<i>hostName</i> <i>hostGroupName</i></pre>	<p>Use the remove volume LUNMapping command to remove volume LUN mapping. Refer to “Remove Volume LUN Mapping” on page 8-54.</p>
<pre>download storageArray file=<i>filename</i> content=firmware [downgrade=(TRUE FALSE)]</pre>	<p>Use the download storageArray firmware command to download firmware. Refer to “Download Storage Array Firmware/NVSRAM” on page 8-43.</p>
<pre>download storageArray file=<i>filename</i> content=NVSRAM</pre>	<p>Use the download storageArray NVSRAM command to download NVSRAM values. Refer to “Download Storage Array NVSRAM” on page 8-44.</p>
<pre>download drive [<i>trayID</i>,<i>slotID</i>] file=<i>filename</i> content=(firmware modePage)</pre>	<p>Use the download storageArray driveFirmware command to download firmware images to all drives in the storage array. Refer to “Download Storage Array Drive Firmware” on page 8-41.</p>
<pre>download (allTrays tray [<i>trayID</i>]) file=<i>filename</i> content=firmware</pre>	<p>Use the download (environmental card) firmware command to download module firmware. Refer to “Download Environmental Card Firmware” on page 8-41.</p>
<pre>download storageArray file=<i>filename</i> content=featureKey</pre>	<p>Use the enable storageArray feature command to enable a feature. Refer to “Enable Storage Array Feature Key” on page 8-45.</p>

Deprecated Command	New Command
remove copyEntry target [targetName] [source [sourceName]]	Use the remove volumeCopy command to remove volume copy entries. Refer to “Recopy Volume Copy” on page 8-46.
remove volumeReservations (allVolumes volume [volumeName])	Use the clear volume command to clear persistent volume reservations. Refer to “Clear Volume Reservations” on page 8-14.
set controller [(a b)] batteryInstallDate=(TRUE FALSE)	Use the reset storageArray batteryInstallDate command to reset the battery date. Refer to “Reset Storage Array Battery Install Date” on page 8-56.
set controller [(a b)] serviceMode=(TRUE FALSE)	Use the set controller availability=serviceMode) command to place the storage array in service mode. Refer to “Set Controller” on page 8-66.
set drive [trayID,slotID] operationalState=(optimal failed)	Use the set drive operationalState=failed command to place the storage array in failed mode. Refer to “Set Drive State” on page 8-69. To return a drive to the optimal state use the revive drive command. Refer to “Revive Drive” on page 8-59.
set performanceMonitor interval=intervalValue iterations=iterationValue	Use the set sessions command to define values for the performance monitor interval and iterations. Refer to “Set Session” on page 8-76.
set storageArray batteryInstallDate=(TRUE FALSE)	Use the reset storageArray batteryInstallDate command to reset the battery date. Refer to “Reset Storage Array Battery Install Date” on page 8-56.
set storageArray clearEventLog=(TRUE FALSE)	Use the clear storageArray eventLog command to clear the storage array event log. Refer to “Clear Storage Array Event Log” on page 8-13.
set storageArray resetConfiguration=(TRUE FALSE)	Use the clear storageArray configuration command to clear the entire configuration from the controllers in a storage array. Refer to “Clear Storage Array Configuration” on page 8-12.

Deprecated Command	New Command
<pre>set storageArray RLSBaseline=currentTime</pre>	Use the <code>reset storageArray RLSBaseline</code> command to resets the RLS baseline for all devices. Refer to “Reset Storage Array RLS Baseline” on page 8-57.
<pre>set storageArray dayOfTime=(TRUE FALSE)</pre>	Use the <code>set storageArray time</code> command to set the clocks on both controllers in a storage array to the clock of the host. Refer to “Set Storage Array Time” on page 8-81.
<pre>set volume [volumeName] mirrorEnabled=(TRUE FALSE)</pre>	Use the <code>set volume</code> command with mirror cache enabled. Refer to “Set Volume” on page 8-82.
<pre>set volumeCopy target [targetName] [source [sourceName]] priority=(lower low medium high highest)</pre>	Use the <code>set volumeCopy</code> command to define the volume copy pair. Refer to “Set Volume Copy” on page 8-87.
<pre>set volumeLabel ID [hexValue] userLabel=volumeName</pre>	Use the <code>set volume</code> command to define a user name for a volume. Refer to “Set Volume” on page 8-82.
<pre>show hostTopology</pre>	Use the <code>show storageArray hostTopology</code> command to show all mappings, storage partition topology, host type labels, and host type index for the host storage array. Refer to “Show Storage Array Host Topology” on page 8-104.
<pre>show volumeReservations (allVolumes volume volumeName]</pre>	Use the <code>show volume reservations</code> command to return information about the volumes that have reservations.. Refer to “Show Volume Reservations” on page 8-116.
<pre>disableSnapshot volume</pre>	Use the <code>stop snapshot</code> command stop a copy-on-write operation. Refer to “Stop Snapshot” on page 8-122.
<pre>recreateSnapshot volume</pre>	Use the <code>recreate snapshot</code> command start a fresh copy-on-write operation using an existing snapshot volume. Refer to “Re-create Snapshot” on page 8-51.

Deprecated Command	New Command
<pre>start increaseVolCapacity volume=<i>volumeName</i> incrementalCapacity=<i>capacityValue</i> drives=(<i>trayID1,slotID1 ... trayIDn,slotIDn</i>)</pre>	Use the <code>set volume</code> command to define values for increasing volume capacity. Refer to “Set Volume” on page 8-82.
<pre>start volumeCopy source="<i>sourceName</i>" target="<i>targetName</i>" copyPriority=(lowest low medium high highest)</pre>	Use the <code>create volumeCopy</code> command to create a volume copy pair and starts the volume copy. Refer to “Create Volume Copy” on page 8-31.
<pre>on error (stop continue)</pre>	Use the <code>set session</code> command to define how the CLI session responds to errors. Refer to “Set Session” on page 8-76.
<pre>use password "<i>storageArrayPassword</i>"</pre>	Use the <code>set session</code> command to define a password for the storage array. Refer to “Set Session” on page 8-76.
<pre>upload storageArray file=<i>filename</i> content=configuration</pre>	Use the <code>save configuration</code> command to save a storage array configuration. Refer to “Save Storage Array Configuration” on page 8-62.
<pre>upload storageArray file=<i>filename</i> content=(allEvents criticalEvents)</pre>	Use the <code>save storageArray (allEvents criticalEvents)</code> command to save a events to a file. Refer to “Save Storage Array Events” on page 8-63.
<pre>upload storageArray file=<i>filename</i> content=performanceStats</pre>	Use the <code>save storageArray performanceStats</code> command to save a the performance statistics to a file. Refer to “Save Storage Array Performance Statistics” on page 8-64.
<pre>upload storageArray file=<i>filename</i> content=RLSCounts</pre>	Use the <code>save storageArray RLSCounts</code> command to save the RLS counters to a file. Refer to “Save Storage Array RLS Counts” on page 8-64.
<pre>upload storageArray file=<i>filename</i> content=stateDump</pre>	Use the <code>save storageArray stateCapture</code> command to save state dumps to a file. Refer to “Save Storage Array State Capture” on page 8-65.

.....

Table C-2 Deprecated Keywords

Old Syntax	New Syntax
usageHint	Removed
enforceSoftLimit	Removed
filesystem	Removed
database	Removed
multimedia	Removed
mapping	lumMapping
featureKey	feature
modePage	Removed
copyEntry	volumeCopy
volumeReservations	reservations
performanceMonitor interval	performanceMonitorInterval
performanceMonitor iterations	performanceMonitorIterations
clearEventLog	clear storageArray eventLog
resetConfiguration	reset storageArray configuration
timeOfDay	time
priority	copyPriority
volumeLabel	Removed
id[]	volume<>
volumeReservations	show volume reservations
disableSnapshot	stop snapshot
recreateSnapshot	recreate snapshot
increaseVolCapacity	set volume addCapacity
incrementalCapacity	addCapacity
on error	set session errorAction
use password	set session password
upload	save

.....

Old Syntax	New Syntax
stateDump	stateCapture

A

- activate
 - Remote Volume Mirroring feature 8-5
 - storage array firmware 8-7
- activating the Remote Volume Mirror 5-9, 5-10, 5-11
- adding comments to a script 2-11
- assigning global hot spares 3-28
- auto configure command, using 3-14
- autoconfigure storage array 8-8
- autoconfigure storage array hot spares 8-10

C

- cache parameters, changing 7-8
- changing
 - cache parameters 7-8
 - controller ownership 7-17
 - RAID levels 7-7
 - Remote Volume Mirror settings 5-14
 - segment size 7-7
 - Snapshot Volume settings 4-11
 - Volume Copy settings 6-11
- channel
 - clear drive channel statistics 8-12
 - set drive channel status 8-68
 - set host channel 8-72
 - start drive channel locate 8-116
 - stop drive channel locate 8-122
- check volume parity 8-11
- clear
 - drive channel statistics 8-12
 - storage array configuration 8-12
 - storage array event log 8-13
 - storage array firmware pending 8-13
 - volume reservations 8-14
 - volume unreadable sectors 8-14
- clearing the configuration 3-13
- CLI commands 1-3
- CLI usage examples 1-13
- clocks, controller, synchronizing 7-5

- collecting physical disk data 7-10
- collection all support data 7-9
- command formatting rules 8-2
- command line interface, how to use 1-3
- command line parameters 1-5
- commands listed alphabetically 8-5
- commands listed by function 8-127
- configuration concepts 3-2
- configuration script example 1 B-2
- configuration script example 2 B-4
- configure
 - autoconfigure storage array 8-8
 - autoconfigure storage array hot spares 8-10
 - changing cache parameters 7-8
 - changing controller ownership 7-17
 - changing RAID levels 7-7
 - changing segment size 7-7
 - create RAID volume (automatic drive select) 8-18
 - create RAID volume (free capacity based select) 8-20
 - create RAID volume (manual drive select) 8-23
 - dynamic volume expansion 5-22
 - storage partitioning 5-21, 6-16
 - using the auto configure command 3-14
- configuring a storage array 3-10
- controller
 - changing controller ownership 7-17
 - controller commands 8-127
 - controllers 3-2
 - diagnose controller 8-37
 - diagnosing a controller 7-11
 - enable controller 8-44
 - enabling controller data transfer 7-4
 - reset controller 8-56
 - resetting a controller 7-4
 - save controller NVSRAM 8-60
 - set controller 8-66
- create
 - host 8-15
 - host group 8-16

.....

- host port 8-17
- RAID volume (automatic drive select) 8-18
- RAID volume (free capacity based select) 8-20
- RAID volume (manual drive select) 8-23
- Remote Volume Mirror 8-26
- Snapshot Volume 8-28
- Volume Copy 8-31

creating

- a Remote Volume Mirror 5-8, 5-12
- a Snapshot Volume 4-4
- a Snapshot Volume by specifying a number or physical disks 4-6
- a Snapshot Volume user assigned physical disks 4-4, 4-5
- a Volume Copy 6-7, 6-8
- volumes in an existing volume group 3-19
- volumes with software assigned physical disks 3-18
- volumes with user assigned physical disks 3-16

.....

D

- data replication 5-3
- deactivate Remote Volume Mirror 8-32
- deactivating the Remote Volume Mirror 5-20
- defragmenting a volume group 7-8
- delete
 - host 8-33
 - host group 8-34
 - host port 8-34
 - volume 8-35
 - volume group 8-36
- deleting a primary or secondary volume 5-18
- detailed error reporting 1-11
- determining
 - candidates for a Remote Volume Mirror pair 5-12
 - Volume Copy candidates 6-7
 - what is on your storage array 3-10
- diagnose
 - controller 8-37
 - Remote Mirror 8-38
- diagnosing a controller 7-11
- disable storage array feature 8-39

- disabling the remote volume mirror 5-19
- download
 - drive firmware 8-39
 - environmental card firmware 8-41
 - storage array drive firmware 8-41
 - storage array firmware NVSRAM 8-43
 - storage array NVSRAM 8-44
- dynamic volume expansion 5-22

.....

E

- enable
 - controller 8-44
 - storage array feature key 8-45
- enabling
 - controller data transfer 7-4
 - the Remote Volume Mirror 5-9
 - the Volume Copy feature 6-7
- examples 8-4
- exit status 1-12

.....

F

- firmware compatibility levels 8-4
- formatting considerations 1-9

.....

H

- host
 - create host 8-15
 - delete host 8-33
 - host topology commands 8-127
 - hosts 3-7
 - set host 8-70
 - set host channel 8-72
 - show storage array host topology 8-104
- host group
 - create host group 8-16
 - delete host group 8-34
 - host groups 3-8
 - set host group 8-72
- host port
 - create host port 8-17
 - delete host port 8-34
 - host port 3-8
 - set host port 8-73

.....

- show host ports 8-97
- hot spare
 - assigning global hot spares 3-28
 - hot spares (about) 3-3
 - set drive hot spare 8-69
- how
 - how Remote Volume Mirroring works 5-2
 - how Snapshot works 4-2
 - how Volume Copy works 6-2

.....

I

- initializing
 - a physical disk 7-17
 - a volume 7-18
- interaction with other features 5-21, 6-16
- interpreting the RLS results 7-15

.....

L

- link interruptions or secondary volume errors 5-4
- locating physical disks 7-5
- logical unit number 3-9

.....

M

- media scan, running 7-2
- mirror relationships 5-3
- mirror repository volumes 5-2
- modifying your configuration 3-21
- module commands 8-130
- monitoring performance 7-6

.....

P

- partitioning, storage 5-21, 6-16
- performance considerations 5-8
- performance tuning 7-6
- persistent reservations, removing 7-4
- physical disks
 - locating 7-5
 - physical disk (drive) commands 8-128
 - physical disks 3-2
- purpose 8-32

.....

R

RAID

- levels 3-6
- levels, changing 7-7
- recover RAID volume 8-47

- read link status diagnostics, running 7-12
- reconstructing a physical disk 7-18
- recopy volume copy 8-46
- re-copying a volume 6-13
- recover RAID volume 8-47
- recovery operations 7-16
- recreate
 - Remote Volume Mirroring repository 8-49
 - Snapshot 8-51
- recurring syntax elements 2-7
- redistributing volumes 7-19
- redundancy check, running 7-3
- Remote Volume Mirror
 - activate Remote Volume Mirroring feature 8-5
 - activating the Remote Volume Mirror feature 5-9, 5-10, 5-11
 - create Remote Volume Mirror 8-26
 - creating a Remote Volume Mirror 5-8, 5-12
 - deactivate Remote Volume Mirror 8-32
 - diagnose Remote Mirror 8-38
 - enabling the Remote Volume Mirror 5-9
 - recreate Remote Volume Mirroring repository 8-49
 - Remote Volume Mirror 6-17
 - Remote Volume Mirror commands 5-5, 8-128
 - remove Remote Volume Mirror 8-52
 - resume Remote Volume Mirror 8-58
 - set Remote Volume Mirror 8-74
 - show Remote Volume Mirror volume candidate 8-98
 - suspend Remote Volume Mirror 8-125
- remove
 - Remote Volume Mirror 8-52
 - Volume Copy 8-53
 - volume LUN mapping 8-54

removing

- a mirror relationship 5-17
- copy pairs 6-15
- persistent reservations 7-4

repair volume parity 8-55

reservations, persistent, removing 7-4

reset

- controller 8-56
- storage array battery install 8-56
- storage array RLS baseline 8-57
- storage array volume distribution 8-57

resetting

- a controller 7-4
- battery age 7-4

restrictions 6-5

resume Remote Volume Mirror 8-58

resynchronization 5-5

revive

- physical disk (drive) 8-59
- volume group 8-60

role reversals 6-18

routine maintenance 7-2

running

- running a media scan 7-2
- running a redundancy check 7-3
- running read link status diagnostics 7-12

S

save

- controller NVSRAM 8-60
- drive log 8-61
- storage array configuration 8-62
- storage array events 8-63
- storage array performance statistics 8-64
- storage array RLS counts 8-64
- storage array state capture 8-65
- storage array support data 8-65

saving a configuration to a file 3-28

script command structure 2-3

script command synopsis 2-5

script command usage guidelines 2-10

session command 8-128

set

- controller 8-66
- drive channel status 8-68
- drive hot spare 8-69

- drive state 8-69
- host 8-70
- host channel 8-72
- host group 8-72
- host port 8-73
- Remote Volume Mirror 8-74
- session 8-76
- Snapshot Volume 8-77
- storage array 8-79
- storage array time 8-81
- storage array tray positions 8-81
- volume 8-82
- Volume Copy 8-87
- volume group 8-88

setting

- controller operational mode 7-16
- modification priority 3-27
- the controller clocks 3-21
- the storage array cache 3-23
- the storage array host 3-22
- the storage array password 3-21

show

- controller 8-89
- controller NVSRAM 8-93, A-17
- drive channel stats 8-95
- drive channel status A-26
- drive download progress 8-96
- host ports 8-97
- physical disk (drive) 8-94, A-20
- Remote Volume Mirror volume
 - candidate 8-98
- storage array 8-99, A-2
- storage array auto configure 8-102
- storage array host topology 8-104
- storage array LUN mappings 8-105
- storage array unreadable sectors 8-107
- string 8-108
- volume 8-108
- volume action progress 8-110
- Volume Copy 8-111
- Volume Copy source candidates 8-112
- Volume Copy target candidates 8-113
- volume group 8-113
- volume performance statistics 8-114
- volume reservations 8-116
- volumes A-27

.....

- snapshot
 - commands 4-3, 8-129
 - Snapshot and repository volume names 4-9
 - volumes 5-21, 6-16
- source volume 6-2
- start
 - drive channel locate 8-116
 - drive initialize 8-117
 - drive locate 8-118
 - drive reconstruction 8-118
 - Remote Volume Mirror
 - synchronization 8-119
 - storage array locate 8-119
 - tray locate 8-120
 - volume group defragment 8-120
 - volume initialization 8-121
- stop
 - drive channel locate 8-122
 - drive locate 8-122
 - Snapshot 8-122
 - storage array drive firmware download 8-123
 - storage array locate 8-123
 - tray locate 8-124
 - Volume Copy 8-124
- stopping
 - Snapshot volume, stopping, restarting, and deleting 4-12
 - Volume Copy 6-14
- storage array
 - commands 8-129
 - performance 6-4
- storage partitioning 5-21, 6-16
- suspend Remote Volume Mirror 8-125
- suspending and resuming a mirror relationship 5-15
- synchronizing controller clocks 7-5

.....

T

- target volume 6-3
- tray loss protection 3-19
- troubleshooting and diagnostics 7-9

.....

U

- uncategorized command 8-130
- user defined parameters 4-7
- using
 - the auto configure command 3-14
 - the create volume command 3-16

.....

V

- viewing Volume Copy properties 6-10
- volume commands 8-130
- Volume Copy
 - commands 6-5, 8-131
 - Volume Copy 5-22
 - Volume Copy and persistent reservations 6-4
- volume group
 - commands 8-131
 - volume groups 3-3
- volumes 3-4

.....

W

- write modes 5-3

.....